Probabilistic Methods for Data Integration in a Multi-Agent Query Answering System

Miklós Nagy

Supervisors: Maria Vargas-Vera, Enrico Motta

KMI-TR-06-08

2006 February

www.kmi.open.ac.uk/publications/papers/kmi-tr-06-08.pdf

Contents:

1. Abstract

This report describes a proposal for a multi agent ontology-mapping framework that makes use of probabilistic information in order to enhance the correctness of the mapping. The proposed research focuses on two correlated areas namely similarity measures with its representation as a Dempster-Shafer belief function and usability of different optimalisation methods for combining these belief functions in a distributed environment. The main goal of our proposed research is to establish a multi agent framework that integrates user query related information from distributed scientific databases utilizing the AQUA system. The outcome of the research will contribute to the feasibility study of a distributed information integration network that is based on the European Commission Joint Research Center's data management and dissemination databases (AlloysDB, GasketDB, CorrosionDB, HTR-FUELDB), which stores mechanical and physical properties of engineering materials produced by the European RTD projects. These databases cover the materials behavior at low, elevated and high temperatures for base materials and welded joints and also includes irradiation materials testing in the field of fusion and fission and thermal barrier coatings tests for gas turbines.

2. Introduction

As envisioned the number of ontologies is growing on the Semantic Web the question of mapping between the concepts described by them received increasing attention by the researcher community. As the different ontology definition languages emerged SHOE[1], RDF(S)[2], DAML[3], OIL[4], DAML+OIL[5] and OWL[6] different methodologies have been proposed to find the correspondent mapping and similarities between two concepts described by different ontologies. The first proposed solutions were mainly build on the logical foundations that gave rise the existence to the particular ontology language. Further solutions proposed the combination of logic and machine learning algorithms that also exploit the textual representation of the content. The first approaches were mainly semi automatic solutions and claimed that fully automatic solutions is hardly possible to imagine considering the fact that the ontologies are the different representations of the human expert's and domain's knowledge. Most recent research on integrated ontology mapping [8] shows that the combination of the different similarity measures proved to provide considerably better results than the individually applied methods, point out that the general problem of dealing with uncertainty inherent to the mapping process has not been a thoroughly investigated area. The main problem with the current approaches that even if a fully automatic mapping is applied and then the inferencing is carried out with incorrect partial results the final result will also be distorted by the errors introduced into the system in the earlier phases.

This research direction attacking the problem of handling uncertainty and reasoning with it in the context of ontology mapping is in its early stages'. Probabilistic-based ontology mapping which is also the main interest of my research is a promising research area that has started to be investigated. Current research has been done to exploit Bayesian Networks [9] to capture and reason about incomplete, partial or uncertain knowledge. Since Dempster-Shafer theory of evidence has more expressive power when it comes to representing total ignorance under uncertainty this forms the main motivating factor of my research.

The organization of the report is as follows:

Chapter 2 describes the motivation, problem description and the expected contribution of our research. It is conceptually part of the formal research proposal, however it is presented separately in the beginning of the document.

Chapter 3 reviews how information and data integration approaches and uncertain reasoning has been investigated in previous literature, and how researchers have defined the ontology-mapping problem in this context. Novel mediation based information and data integration approaches are then presented and contrasted with earlier work, before a short analysis of the previous work. It is clearly recognised that that modeling of uncertainty in the context of ontology mapping is a getting more attention however there is a gap in research in the context of Information and data integration.

Chapter 4 addresses specific outputs of the research to date. In particular the proposed system architecture is discussed with respect to the particular ontologies and queries that will serve as a test bed in our implementation. It also introduces the similarity and uncertainty issues that need to be investigated in our future research.

Chapter 5 details the questions that will be addressed by the research. The methods that will be used are also discussed including overall plans for how the research will be carried out.

Chapter 6 presents a list of references used in the literature review and Chapter 7 shows ontology fragments as well as a snapshot of our development environment.

2.1 Motivation:

The recent popularity of the Web created a demand for software solutions that help the user to sort out relevant information from the vast number of data available in this media represented by static, dynamic web pages and Web enabled databases. Besides the popular search engines like (Yahoo, Google) a

very promising solution is the question-answering (QA) system, which provides precise answers to specific questions raised by the user. A very advanced system called AQUA[43], which amalgamates Natural Language Processing (NLP), Logic, Ontologies and Information retrieval techniques is envisioned to play an important role in the development towards the Semantic Web. Considering the dynamic nature of the Semantic web that is the extension of the current World Wide Web (WWW) it is hardly imaginable that isolated applications will be able to serve successfully the users' ever growing requirements since the information available to human decision makers continues to grow beyond human cognitive capabilities. In such an environment a single agent or application limited by its knowledge, perspective and its computational resources cannot cope with the before mentioned scenarios effectively. As the domain becomes larger and more complex, open, and distributed, a set of cooperating agents is needed to address the reasoning task effectively.

Each agent carries only a partial knowledge representation about the domain and can observe the domain from a partial perspective where available prior knowledge is generally uncertain. Extensive study of the subtask of how multiple agents can collectively reason about the state of the domain based on their local knowledge, local observation, and limited communication needs to be carried out in order to ensure that an agent in a multi-agent system can reason and act autonomously as in the single-agent paradigm, to overcome its limit in domain knowledge, perspective, and computational resource, it can benefit from other agents' knowledge, perspectives, and computational resources through communication and coordination. Within the AQUA query answering approach a complex question being asked by the user, a broker agent then divides the user's question into sub questions, which are passed to specialist agents. Each agent is capable of querying a potential resource. When the specialist agents return the result back to the broker agent, the broker composes a single coherent answer to the user's original question. In the above mentioned multi agent architecture where different agents have access to different heterogeneous, distributed sources and these agents are responsible for a domain specific area. As part of a

query answering system like AQUA, before answering the posed query, agents need to establish mapping between their concepts and properties in their domain specific ontologies in order to provide meaningful and integrated information that corresponds to the query. In the context of the Semantic Web mapping between concepts and their relations to each other needs to be established on the fly instead of using a mediated ontology that was created beforehand. This implies that agents engage in negotiating about the concepts and their relationships that is present in their different ontologies in order to integrate the data. Building such mapping involves reasoning under uncertainty about the similarity of concepts in the different ontologies. I intend to investigate the Dempster-Shafer theory as an uncertainty representing and reasoning framework because one of the advantages of the framework is that priors and conditionals need not be specified, unlike Bayesian methods which often use symmetry arguments to assign prior probabilities to random variables (e.g. assigning 0.5 to binary values in which no information is available). Further advantages as compared with e.g. fuzzy logic or Bayesian theory is that it allows the user to represent uncertainty in the knowledge representation, because the interval between support and plausibility can be easily assessed for a set of hypothesizes. Missing data also could be modeled by Dempster-Shafer approach and additionally evidences from two or more sources can be combined using Dempster's rule of combination. The combined support, plausibility, disbelief, and uncertainty can each be separately evaluated. Historically the applicability of the Dempster-Shafer theory was extensively examined in the context of expert systems, which often deals with multiple mutually exclusive hypotheses. One example is GERTIS [57] system, which uses taxonomic structure of the hypothesis space to present pieces of evidence relevant to a diagnosis in an order that reflects the experts' reasoning process. GERTIS uses Dempster-Shafer-based reasoning model for diagnosing hierarchically related hypotheses, but also suggests ways to generate better explanations by using knowledge about the structure of the hypothesis space. One observation made about this kind of problem domain is that the set of hypotheses that are of interest to human expert often form a taxonomic class

hierarchy where the leaf nodes represent single hypotheses, and an internal node represents the union of its children nodes. This kind of hypotheses thus forms a hierarchical hypothesis space. The task of combining evidence bearing on hierarchical hypotheses is complicated by the impreciseness of the evidential strengths because evidence could bear on hypotheses in higher levels of the hierarchy, but gives no further information about the relative likelihood of their subclasses. Knowledge represented by ontologies (set of hierarchical concepts) is clearly a similar scenario and we believe that it is worth to investigate the practical applicability of the Dempster-Shafer theory in our ontology-mapping context. Further the Dempster-Shafer theory offers a promising alternative to traditional uncertainty handling formalisms such as the Bayesian theorem because it captures the impreciseness of evidential strengths by allowing them to bear on sets of hypotheses directly.

2.2 Problem definition

I classify the problem domain into two distinctive but correlated areas:

- Semantic mapping generation algorithms between ontologies have an inherent drawbacks when it comes to integrating data and information in real life scenario like question answering. Effectiveness can be improved but efficiency will worsen and vice versa. The reason for this is mapping process requires considerable background knowledge about the concepts, properties and its relation on the domain and different mapping algorithms use different information (e.g. predefined rules, machine learning) to assess similarity in advance instead of in real time. The problem is when mapping is made a priori, our real time question answering system will likely to fail when the resource (domain ontology, source) changes in the dynamic Web environment.

  Research questions that need to be answered are:

- How to create mapping algorithms in a distributed environment that both effective and efficient and comparable to traditional solutions?
- How to replace the global knowledge with a distributed local knowledge of the multi agent system effectively?
- How trust in the different source information can be harnessed during the similarity combination process?

- Uncertainty handling requires human expert involvement in order to improve effectiveness of the system. Since uncertainty involves computing and combining probability distributions for all possible events any uncertainty handling formalisms are computationally expensive operations. Dempster-Shafer theory of evidence provides a promising alternative for reasoning under uncertainty. However the applicability in practical, real life scenarios is limited by the fact that combining the pieces of evidence with the Dempster's rule of combinations suffers from the exponential growth of the state space therefore any system could be infeasible to built even with relatively small number of variables. Multi-variate Dempster-Shafer just worsen the situation since in this scenario each variable can have multiply values so the number of possible focal set is much bigger. Illustrating the problem consider that ϕ has a domain of $D = \{x_1, ..., x_i\}$ and each variable x_i has n configuration. The size of the state space [36] is 3^n.

$\lvert D \rvert$	n	2^n	3^n
1	2	4	9
2	4	16	81
3	8	256	6561
4	16	65536	43046721
5	32	4294967296	1853020188851841

Advanced optimalizations and approximation helps to decrease the size of the state space and make the combination feasible although the applicability and feasibility of these methods and architectures have not been investigated in a distributed environment. Research questions that need to be answered are:

- How Dempster-Shafer theory of evidence can be applied in a distributed multi agent environment for large complex domains?
- How mass probability can automatically be assessed from the similarity algorithms by specialised agents?
- How to apply traditional optimalization techniques in a distributed environment where pieces of the evidence are distributed between agents?

2.3 Expected contributions

Ontology mapping is widely investigated area and a numerous approaches led to different solutions. To date uncertainty handling during the mapping process was not in the focus of the research community since initially only different logic(FOL,DL) based approaches has been utilized. As practical application of ontologies emerged on the web it has been acknowledged that considering the dynamic nature of the Web the problem of inconsistencies, controversies and lack of information needs to be handled. First systems that used probabilistic information like LSD, GLUE proved that combining different similarity measures based on their probability could significantly improve the accuracy of the mapping process. I believe that probability theory and distribution does not have enough expressive power to tackle certain aspects of the uncertainty e.g. total ignorance. In order to solve the before mentioned problem I chose Dempster-Shafer theory as a formalism for representing uncertainty. To justify my choice the following requirements were identified:

- Conditional and a-priori probability cannot be assessed for all problem sets.

- Probability values are assigned to sets of possibilities rather than single events.
- Due to lack of information the ignorance needs to be represented.

As a consequence I think evidence (Dempster-Shafer) theory is the most suitable approach and needs to be investigated in ontology mapping context thought this has not been done so far. The reason is that Dempster Shafer combination rule can easily be unfeasible in case of domains with large number of variables. Different optimalisations methods have been developed but to date I could not find approaches that considered distributed environment. Local computation and valuation networks uses joint tree structure to narrow down the number of focal elements and different architectures has been proposed based on message passing schemes to carry our inference and resolve the problem of the Dempser's rule of combination. In my scenario I assume a dynamic multi agent environment where different agents has partial knowledge of the domain. I believe that valuation network is a prosperous candidate for my scenario however the problem of distributed knowledge and inference with its implications e.g. distributed joint tree construction needs to be addressed in my PhD research.

The expected contributions of my Ph.D. research can be grouped into two main areas:

1. Ontology mapping with multi agent for system integration in the context of Query answering:
    - Establishing an effective ontology concept similarity measures and combination algorithms based on concept name, property hierarchy structure and instance values in a distributed environment.
    - Assessing and incorporating trustworthiness of the sources into the combination rule based on domain, author and time related information.

- Agent communication strategies for combining pieces of evidence (Dempster-Shafer) where the evidence is distributed among specialized agents.

2. Reasoning under uncertainty using Dempster Shafer theory of evidence.
   - Algorithms for determining the variable elimination sequence when the joint tree is distributed between the agents.
   - Algorithms for building up a joint tree in the distributed environment.

3. Literature review

3.1 Information integration approaches

During the past decades the different research communities have investigated the information integration problem that lead to numerous different approaches in a way in which different information sources can be integrated.

Derived from the data engineering community several solutions have been proposed that based on a mediator architecture where logical database schemas are used as shared mediated views over the queried schemas. A number of systems have been proposed e.g. TSIMMIS[13], Information Manifold [14], InfoSleuth [15], MOMIS [16] that shows the flexibility and the scalability of these approaches.

Derived from the knowledge engineering community solutions the use of ontologies (conceptual domain knowledge schemas) is the main approach for resolving semantic differences in heterogeneous data sources. Based on this approach several sub categories can be identified:

- *Creating a global ontology:* all the different sources share the same ontology in order to make information integration possible. These solutions fit well when the number of sources is limited and a consensus can be achieved between partners. Based on the real life scenarios this solution is really inflexible in nature and does not considered as a viable alternative in the context of the WWW environment.
- *Ontology merging*: semantic integration is achieved through merging the different source ontologies into a consistent union of the source ontologies. These systems make use of the fact that different ontologies have overlapping fragments that is the basis of merging process. FCA-MERGE [17] uses bottom-up approach with structural description of the merging process and applies techniques from natural language processing and formal concept analysis. PROMPT [18] is a semi

automatic ontology merging tool that makes initial suggestions based on linguistic similarity between class names then performs automatic updates, find new conflicts and makes new suggestions.

- *Ontology mapping*: semantic integration is achieved through creating mappings between concepts, attributes etc. between two ontology entities. A wide range of techniques has been proposed from the manually defined rules to the semi automatic approaches that make use of machine learning, heuristics, natural language processing and graph matching algorithms. MAFRA [19] a mapping framework for distributed ontologies supports interactive, incremental and dynamic ontology mapping process in the Semantic Web context. It's main contribution is that it creates a true distributed ontology mapping framework that is differ from the mediator based approaches. GLUE [11] evolved from a mediator based LSD data source schema matching, applies machine learning techniques and similarity measures based on joint probabilistic distributions.

InfoSleuth[15] figure 1 defines a set of agents organized in different conceptual layers with different tasks to collect data to create information for higher level of abstraction. Different agents from different levels of abstraction interact to resolve complex problems during information routing, extraction, analysis and integration. Based on their system agents are fall into three categories:

1. User agents: main task is interface between the users and the system, typically transforms user queries into a form that can be feed to the system.
2. Resource agents: wrap the heterogeneous sources into a common format that the system can process. The sources can be diverse from flat files to databases.
3. Core agents: gather, analyze, and filter information to process the user request. These agents can be further classified into:

- Service agents: provide internal information to the system, include broker agents that maintain the knowledge base of the agents, ontology agents maintain the knowledge base of the different ontologies, monitor agents that monitor the operation of the system.
- Query and analysis agents: fuse/analyze information from different sources. Their main task is to process complex queries that span multiple resources.
- Planning and temporal agents: guides the requests though processing. Their main function is to plan how the user request should be processed.

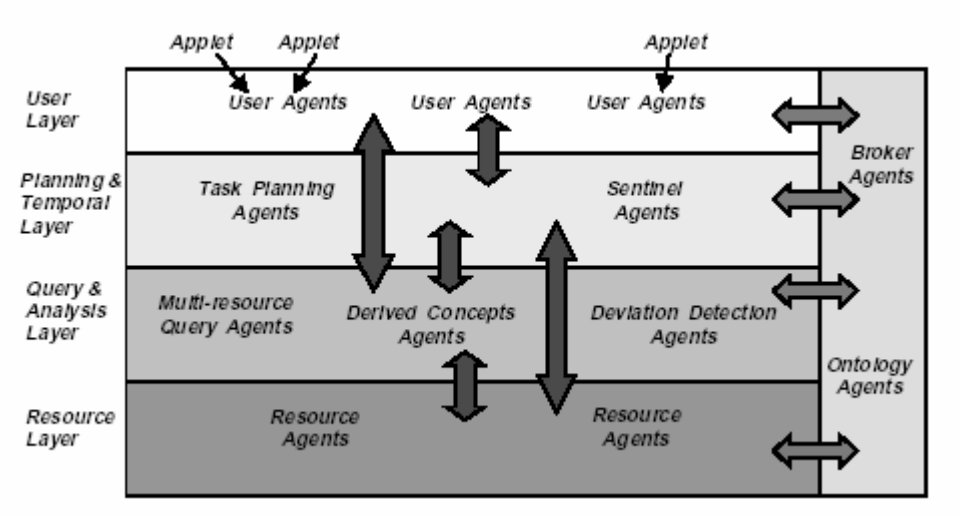


Figure 1.

Real application domain experiments with InfoSleuth agent based system for information gathering concludes that a real information gathering application requires goal-driven interaction between information access, integration and analysis.

MOMIS [16] figure 2 is based on the ARPA I^3 (coordination, mediation and wrapping service) architecture and integrates data from traditional databases (relational, object oriental) to semi-structured data. The system adopts the

common data model (ODM) and language (ODL) describing information integration regarding the sources, which support inference and reasoning in source integration. The system utilizes hierarchical clustering techniques with the ARTEMIS support tool in order to provide support for semi-automatic, schema based integration process.

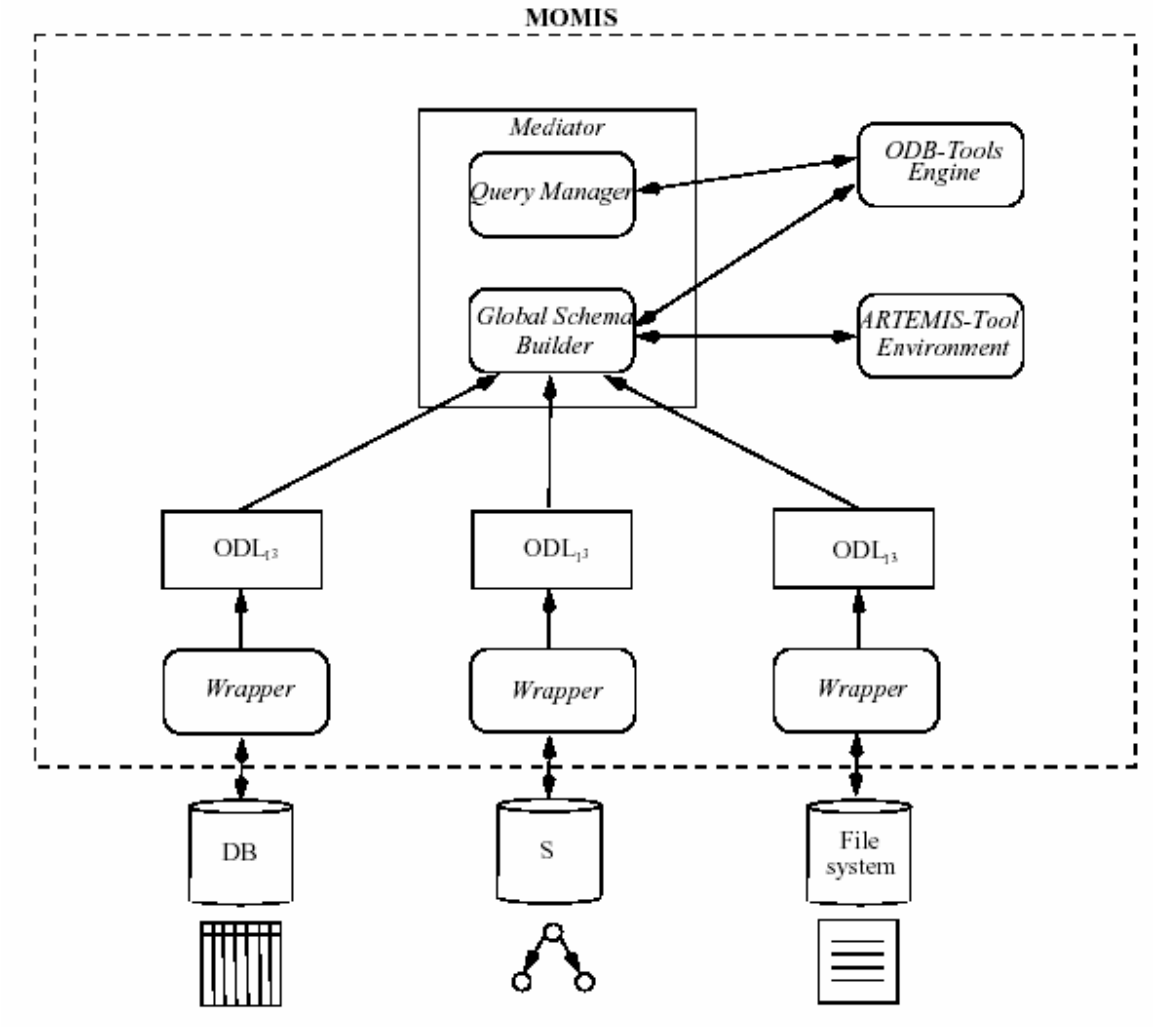


Figure 2.

The major components of the system are:

1. Wrappers: responsible to translate the schema of the data source into ODL language and the translation of the query expressed in ODL into the local data source format that can be processed locally.

2. Mediator consist of two modules:
    - Global schema builder (GSB) combine, integrate and refine data coming from source wrappers.
    - Query manager (QM) performs query processing and optimalization based on description logic. It can translate the given submitted query into sub queries relevant to the local sources.
3. ODB tools engine: based on OLCD description logic performs schema validation and query optimalisation.
4. Artemis tool environment: performs schema analysis and clustering.

The integration process is based on the following steps:

1. Extraction of terminological relationships: a common thesaurus is constructed which expresses inter-schema knowledge among different sources. Terminological relationships are derived semi-automatically from ODL schema descriptions.
2. Clustering of ODL classes: relation between ODL classes are evaluated with measuring the relationships between ODL classes based on their names and attributes which is grouped together using hierarchical clustering techniques.
3. Mediator (global schema) construction: the created clusters of ODL classes are analyzed in order to construct a global schema. An integrated ODL class defined for each cluster is describe the cluster and characterized by the union of their attributes.

The MOMIS system relies on integration knowledge that includes local source schema data, virtual mediated schema and it's mappings what is given in terms of description logic.

3.2 Integrated ontology mapping approaches

Derived from the SWAP[10] (Semantic Web and Peer-to-Peer) project an integrated ontology mapping[8] approach has been proposed where ontologies represent a local view of individual resources such as e-mails, bookmarks etc. in a peer to peer network. To answer natural language queries the system needs to map up between the different concepts kept in the local ontologies. The assumption of the approach is that since any metadata represented by ontologies, which is easily interpretable, additional knowledge can be deducted in order to measure similarities between the different entities. The authors define the mapping and similarity as follows:

$map : O_{i1} \rightarrow O_{i2}$ defined as $map(e_{i1j1}) = e_{i2j2}$ if $sim(e_{i1j1}, e_{i2j2}) > t$ where t is a certain threshold and e_{ij} are entities.

To achieve mapping that is based on a set of rules where each rule suggest a hint for the similarity and the result of the set of rules will provide information for making the final decision about the similarity of the particular entities. The suggested approach defines certain similarity measurements (figure. 3), which are used during the mapping process

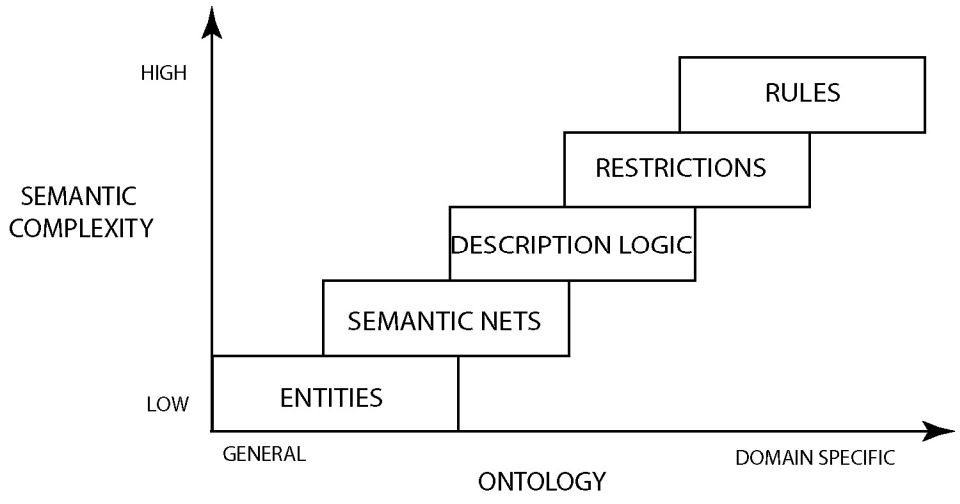


Figure 3.

To prove the hypothesis that the combination of the partial results provides better similarity measure authors utilized and compared the following methods to integrate the similarities measures:

- Weighted sum: $sim(e_{i1j1}, e_{i2j2}) = \sum_{k-1}^{n} w_k sim_k(e_{i1j1}, e_{i2j2})$ where w_k is a given weight of the sim_k
- Sigmoid function: $sim(e_{i1j1}, e_{i2j2}) = \sum_{k-1}^{n} w_k \times sig_k(sim_k(e_{i1j1}, e_{i2j2}) - 0.5)$ where $sig(x) = \frac{1}{1+e^{-ax}}$ and w_k is a given weight of the sim_k
- Machine learning with neural networks: three layer neural network with a linear input layer, hidden layer with *tanh* function and a *sigmoid* output function.

As a conclusion authors state that based on experiments an average of 20 % precision gain can be achieved but as it is also pointed out in the model, uncertainty and inexact nature of the applied similarity measures are not handled and this inherent general problem can bring down the precision results.

The GLUE [11] (figure 4) system that has been has been evolved from the LSD [12] (Learning source descriptions) schema matching research project use multi strategy learning for computing the concept similarities. Figure 2 shows the GLUE system architecture. According to the proposed model the similarity is defined by the joint probability distribution between two concepts where each concept is from the subset of instances taken from a finite universe of instances. This distribution describes four probabilities $P(A,B), P(A,\overline{B}), P(\overline{A},B), P(\overline{A},\overline{B})$ where each describes that the probability of a randomly chosen instance belongs to A and not to B. Based in this representation many similarity measures can be expressed. In the experiments authors mostly use:

$$Jaccard - sim(A,B) = \frac{P(A \cap B)}{P(A \cup B)} = \frac{P(A,B)}{P(A,B) + P(A,\overline{B}) + P(\overline{A},B)}$$ where the value 0 represent the disjoint between A and B and 1 means A and B are the same concepts.

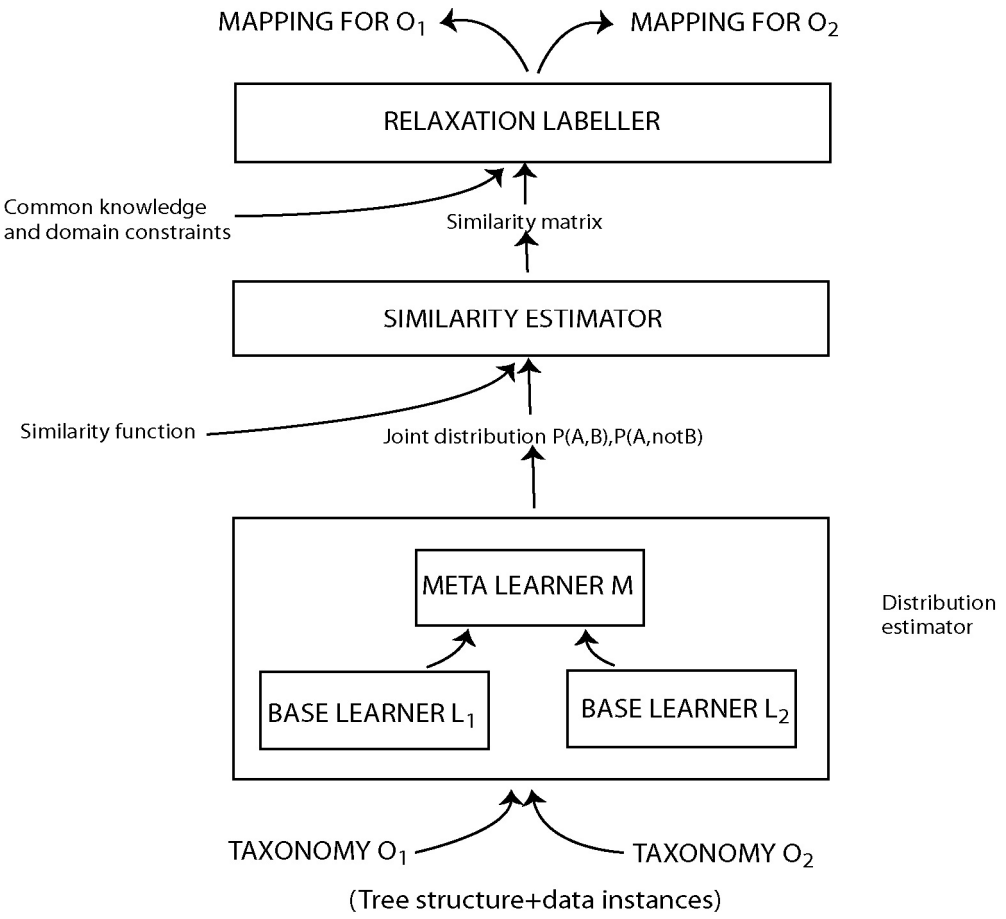


Figure 4.

The set of meta and the base learners which make up the Distribution Estimator operate on the two ontologies and compute the joint probability for every pair of concepts ($A \in O_1, B \in O_2$). The role of the relaxation labeler is that based on the similarity matrix, heuristics, local knowledge and domain specific constraints it searches for mapping configuration that best satisfies the requirements. The current implementation contains two kind of learners:

1. Content learner: exploits the frequencies of words in a textual content of an instance employing naïve bayes learning technique.
2. Name learner: makes predictions using the name of the instance instead of the content.
3. Meta learner: combines the results of the base learners using weighting each similarity.

Worth to note that this kind of weighting mechanism represents some sort of trust in the particular result. Finally based on the similarities using relaxation labeling the labels are assigned to the nodes of the graph with the following formula:

$P(X = L\Delta_k) \propto \sum_{M_x} \delta(\sum_{k=1}^{n} \alpha_k f_k(M_x, \Delta_x, X, L) \times \prod_{(x_i = L_i) \in M_x} P(x_i = L_i \mid \Delta k)$ where X, L are the nodes in the taxonomy, Δ_k all information about the domain, M_x is all possible label assignment, $f_k(M_x, \Delta_x, X, L)$ is a feature function, $\propto$ is a proportional operator and α indicates the importance of feature $f_k()$. δ is the sigmoid function that describes the linear combination of the feature functions to estimate probability. The result if the formula describes the evidence of the nodes match.

Based on experiment the matching accuracy is between 66-97% when the result of the base learners are combined in contrast to the 52-83% of the content learner's and 12-15% of the name learner's performance.

The GLUE approach of the ontology-matching problem boasts of impressive accuracy results, but nevertheless the methodology relies heavily on the existence of the instances that need to be present in order to apply the learners successfully. Another remark is that it uses some kind of evidence or trust during the mapping the similarity estimator could not handle well concepts that are ambiguous.

3.3 Trust in multi agent systems

In general trust is a way for social beings to cope with the uncertainty they face in everyday life. In the context of software agents trust indicates if the communicated information is trustworthy enough for whatever criteria is important to that communication. A variety of factors influences trust establishment, among them integrity, reputation, credibility, reliability, congruity, predictablity, and responsibility. The decision whether or not to trust a piece of information can depend on many contextual factors e.g. the source of the data, the location and the context of evaluator.
There are various risks to agent systems that require assumption of trustworthiness when agents may encounter that other agents are unreliable in a particular scenario. The benefits of trust in these scenarios are twofold:

1. Allow agents to determine whether other agents are trustworthy or not.
2. Allow agents to assign different trust levels to agents in different situations.

Further dependencies between trust and similarity should be examined in our research in order to increase the accuracy of the recommended similarity measure and to outperform both a simple combination method and any other algorithm. We believe that the trustworthier a similarity measure is the higher will be its impact on the combined result.
Generally, context information can characterize a situation of any similarity measure that is relevant to the interaction between the agents. However, in order to provide meaningful results, one should make use of trust to single out relevant and reliable information sources from unreliable one.

Trust is a fundamental concern in large-scale open distributed systems. It is a fundamental concept for all interactions between the entities that have to operate in uncertain and constantly changing environments. Agents face significant

degrees of uncertainty in making decisions (i.e. it can be hard or impossible to devise probabilities for events happening). In such circumstances, agents have to trust each other in order to minimise the uncertainty associated with interactions. Trust has been defined in a number of ways [60] in different domains. In a multi agent system we adopt the conceptualisation levels of trust [61] in the following ways:

- individual-level trust, whereby an agent has some beliefs about the honesty
- nature of its interaction partners;
- system-level trust, whereby the actors in the system are forced to be trustworthy by the rules of
- encounter (i.e. protocols and mechanisms) that regulate the system.

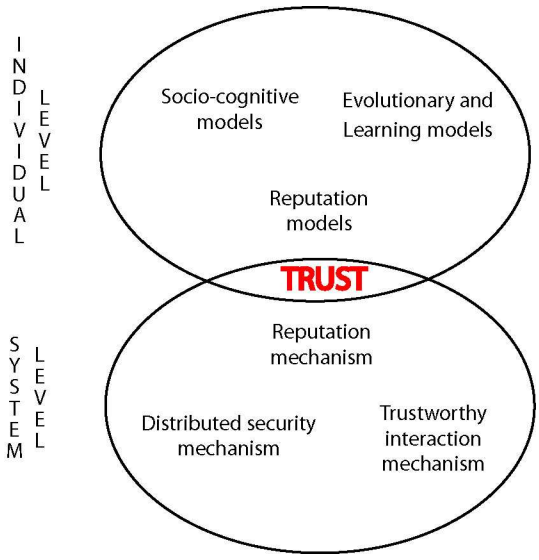


Figure 5. Trust models in multi agent systems

These approaches are also complementary (figure 5) to each other. While protocols aim to ensure the trustworthiness of agents at the system level, they cannot always achieve this objective without some loss in efficiency, and, in such cases, trust models at the individual level are important in guiding an agent's decision making. Similarly, where trust models at the individual level cannot cope

with the overwhelming uncertainty in the environment, system-level trust models, through certain mechanisms, aim to constrain the interaction and reduce this uncertainty.

3.3.1 Individual-level trust

We consider individual level trust when an agent situated in an open environment trying to choose the most reliable interaction partner from a pool of potential agents and deciding on the strategy to adopt with it (i.e. the who, when, and how of interactions). One possible scenario e.g. when agents interact with each other and learn their behavior over a number of encounters so the most reliable agents from the pool can be singled out from the less reliable ones. In this case, the agent reasons about the outcome of the direct interactions with others and could ask other agents about their perception of the potential partners. If sufficient information is obtained and if this information can be trusted, the agent can reliably choose its interaction partners. Further agents could form coherent beliefs about different characteristics of these agents and reasoning about these beliefs in order to decide how much trust should be put in them. As a consequence trust models at the individual level are either learning (and evolution) based, reputation based, or socio-cognitive based.

3.3.1.1 Learning and evolving trust

When agents repeatedly interact with each other any break down in an interaction could mean that the agent does not reliable enough and the possibility of future interactions may need to be avoided. Fruitful cooperation between agents would lead to a higher payoff for both parties. This kind of reasoning is adapted from game theory [62] where all interactions are considered as games with different payoffs (e.g. winning or losing the game) for the individual players

(i.e. the interaction partners). In such games, the safest (i.e. minimising possible loss), and not necessarily the most profitable, move will be chosen unless there can be some way to ascertain that the other party can be trusted. Thus, if an agent believes its counterpart is reciprocative, then the former will never defect, otherwise it will, and both could end up with lower payoffs than if they trusted each other or learnt to trust each other. This belief may only be acquired if the game is repeated a number of times such that there is an opportunity for the agents to learn their opponent's strategy or adapt to each other's strategy.

3.3.1.2 Reputation models

Reputation can be understood as the aggregation of opinions of members of the community about one of them [63]. In multi-agent systems, reputation can be useful when there are a large number of agents interacting with each other. Most reputation models stems from the concept of a social network from sociology [64], which assumes that agents are related to each other whenever they have roles that interconnect them or whenever they have communication links established between one another. Through this network of social relationships, it is assumed that agents can transmit information about each other. There are several aspects of reputation such as:

- gather ratings that define the trustworthiness of an agent, using relationships
- existing between members of the community
- reasoning methods that gathers as much information from the aggregation of ratings retrieved from the community
- promote ratings that truly describe the trustworthiness of an agent

3.3.1.3 Socio-cognitive models of trust

This model assumes trustworthiness of an opponent through subjective perception. Such kind of beliefs are normally stored in an agent's mental state

and are essential in assessing an agent's reliability in doing what they are capable of, or their willingness to do what they say. These models highlight the importance of a cognitive view of trust e.g. Belief–Desire–Intention agents [65] in contrast to a mere quantitative view of trust and define different kind of belief such as:

- Competence belief: a positive evaluation of agent "A" by agent "B" saying that agent "B" is capable of carrying out the delegated task as expected
- Willingness belief: agent "A" believes that agent "B" has decided and intends to do what they have proposed to do
- Persistence belief: agent "A" believes that agent "B" is stable enough about their intention to do what they have proposed to do
- Motivation belief: agent "A" believes that agent "B" has some motives to help agent "A", and that these motives will probably prevail over other motives negative to agent "A" in case of conflict

3.3.2 System-level trust

In the context of open multi-agent systems, interacting agents can utilize a number of mechanisms or protocols that determine the rules of trust. These rules enable an agent to trust other agents based on different constraints imposed by the system e.g. agent's reputation as being a untruthful can be spread by the system or agents can be screened upon entering the system by providing proof of their reliability through the references of a trusted third party. As a consequence trust mechanisms on system-level are either trustworthy interaction protocols, reputation mechanisms that foster trustworthy behavior or other distributed security mechanisms that ensure agents can be trusted.

3.3.2.1 Trustworthy interaction protocols

In order to ensure truth telling of the agents involved in an interaction, a number of protocols and mechanisms have been developed [66] in recent years, which

prevents agents from lying or speculating during interaction. Such protocols ensure that agents have no better option than communicating the truth. In order to avoid malicious agents to exploit the trustworthy environment agents can share their ratings of their opponent with other agents once they have interacted with them.

3.3.2.2 Reputation mechanisms

Reputation mechanisms can operate through centralised or distributed entities that store ratings provided by agents about their interaction partners and then publicise these ratings, such that all agents in the environment have access to them. In this case, it is the system that manages the aggregation and retrieval of ratings as opposed to reputation models, which leave the task to the agents themselves. Reputation mechanisms aim to induce truthful ratings from witnesses and actually make it rational for agents to give ratings about each other to the system.

3.3.2.3 Distributed security mechanisms

In the domain of network security, trust is used to describe the fact that a user can prove who they say they are. This normally implies that agents can be authenticated by trusted third parties those that can be relied upon to be trustworthy and as such are authorities in the system.
A number of security requirements [67] have been proposed that are essential for agents to trust each other and each other's messages transmitted across the network. These include:

- Identity: the ability to determine the identity of an entity. This may include the ability to determine the identity of the owner of an agent
- Access permissions: the ability to determine what access rights must be given to an agent in the system, based on the identity of the agent

- Content integrity: the ability to determine whether a piece of software, a message, or other data has been modified since it has been dispatched by its originating source
- Content privacy: the ability to ensure that only the designated identities can examine a message or other data. To the others, the information is obscured

3.4 Approaches to probabilistic reasoning under uncertainty

The Baysian networks and different variants dominate current research addressing the qualitative reasoning and decision-making problem under uncertainty. Although these approaches successfully lead to numerous real world applications there are several situations where the problem cannot be represented properly within the classical probability framework. These situation include:

- Total ignorance need to be represented
- A-priory probabilistic distributions cannot be fully constructed from the available information.
- Probability mass needs to be assessed to sets or intervals

Three major frameworks that satisfies the above mentioned requirements has been investigated by the researchers [41,42] namely:

- Imprecise probabilities
- Possibility theory
- Dempster-Shafer theory of evidence

As a consequence of these frameworks the following implications are identified which involves situations where there is little available information on which probability can be evaluated or the information is non-specific or conflicting:

1. It is not necessary to determine precise probabilities from an expert or an experiment if it is not feasible to do so.
2. Applying the Principle of Insufficient Reasoning can be avoided

3. Additivity axiom is not imposed

The applicability of the above mentioned theories is still an active research topic it can be noted as a fact that comparing the other theories the Dempster-Shafer theory has a relatively high degree of theoretical development, has relation to the traditional probability theory and set theory. Because of it embodies versatility when it comes to representing and combining different types of evidence from multiply sources a large number of practical applications can be found in the literature.

3.4.1 Dempster-Shafer Theory of evidence

The Dempster-Shafer theory, which provides a mechanism for modeling and reasoning with uncertain information in a numerical way especially when it is not possible to assign a belief to a single element of a set of values, has been introduced by Shafer [21] based on the seminal work of Dempster [20] gives an alternative approach to Bayesian networks and fuzzy sets to represent uncertainty. The theory is based on two ideas:

1. Obtaining a degree of belief from subjective probabilities.
2. Dempster's rule for combining such belief when they are based on independent items of evidence.

Belief results from uncertainty what is usually quantified by probability functions.
In Dempster's model the belief functions are defined as multi-valued mappings where a probability distribution P_A exists on A. The main advantage of the dempster shafer (D-S) theory over the classical probabilistic theories that the evidence of different levels of abstraction can be represented in a way that clear discrimination can be made between uncertainty and ignorance. Further advantage is that the theory provides a method for combining the effect of different learned evidences to a new belief by the means of the Dempster's combination rule.

Due to the fact that a great number of models and justification can be found in the literature the Dempster-Shafer theory has not been applied for concrete problems as extensively as e.g. Bayesian networks or fuzzy sets. Since any model that deals with belief can characterized by two subcomponents:

1. Static part that describes the state of belief. Most of the justification and models based on D-S agree on static part.
2. Dynamic part that describes how the belief needs to be updated if a new evidence has been learned. Originality of models and justifications based on belief function comes from the dynamic part.

Static component of the D-S theory:

Open and closed world: The difference is where the truth lies. In case of a closed world it has to be in the frame of discernment and in case of open world it can be elsewhere.

Frame of Discernment (Θ): finite set representing the space of hypothesizes. It contains all possible mutually exclusive context events of the same kind.

Evidence: available certain fact and is usually a result of observation. Used during the reasoning process to choose the best hypothesis in Θ.

Belief mass function (m): is a finite amount of support assigned to the subset of Θ. It represents a strength of some evidence and

$$\sum_{A \subseteq \Theta} m(A) = 1$$

where m(A) is our exact belief in a proposition represented by A.

Belief: amount of justified support to A that is the lower probability function of Dempster which accounts for all evidence E_k that supports the given proposition A.

$$belief_i(A) = \sum_{E_k \subseteq A} m_i(E_k)$$

Plausibility: amount of potential support on A that is the upper probability function of Dempster, which accounts for all the observations that do not rule out the given proposition.

$$plausibility_i(A) = 1 - \sum_{E_k \cap A = \varnothing} m_i(E_k)$$

Ignorance: the lack of information.

$$ignorance(A) = plausibility(A) - belief(A)$$

Doubt: measure of support, which will never be assigned to A.

$$doubt(A) = 1 - plausibility(A) = belief(\overline{A})$$

Dynamic component of the D-S theory:

Suppose we have two mass functions $m_i(E_k)$ and $m_j(E_k^{'})$ and we want to combine them into a global $m_{ij}(A)$. Following Dempster's combination rule

$$m_{ij}(A) = m_i \oplus m_j = \sum_{E_k \cap E_{k'}} m_i(E_k) * m_j(E_{k'})$$

However when $E_k \cap E_{k='} = \varnothing$ the mass $m_i(E_k) * m_j(E_{k'})$ would go to $\varnothing$, it is necessary to normalize the mass function with the lost mass so

$$m_{ij}(A) = \frac{\sum_{E_k \cap E_{k'}} m_i(E_k) * m_j(E_{k'})}{1 - \sum_{E_k \cap E_{k'} = \varnothing} m_i(E_k) * m_j(E_{k'})}$$

3.5 Approaches to advanced optimalizations and approximations

Based on the Dempster's rule of combination the normalized belief can be computed by producing the joint potential with combining each of the independent potentials. This solution can become quickly infeasible if the number of variables is high since the frame of discernment is represented by a state space with the size of 2^n were *n* is the cardinality of Θ. To overcome this problem a number of methods have been proposed:

1. Approximation: these methods involve biases on theoretical base, some of them result in the loss of important properties. The main principle in these methods is to remove the focal element and redistribute the numerical

values. Several theoretical studies exists but the more relevant are the bayesian [25], summarization[26], k-1x[27],D1[28] approximations.

2. Probabilistic methods: these approaches presume that the initial belief masses are assessed by domain experts, so information sources does not involve high level of doubt what is represented as a reliability factor. Based on this reliability Monte Carlo methods [29,30,31] has been proposed to compute D-S like combinations.
3. Fast moebius transformation: The principle is that combinations are carried out with Fourier transformation where the Moebius transformation can be represented as generalized Fourier transformation [31].
4. Hypertree optimalisations: The principle is belief functions transfer can be modeled with local message passing where the belief function is propagated through the network represented by a hyper tree or Markov tree. D-S system can be modeled by hypergraphs[35] where the potentials $P_1..P_n$ are defined on domains $d_1..d_n$. The combination sequence for $P_1..P_n$ determines the hypertree for the hypergraph where several covering hypertree exist, which can lead to several combination sequences. Considering that the Dempster's rule of combination is commutitative that is :

$$P_1 \otimes P_2 \otimes .. \otimes P_n = P_n \otimes P_1 \otimes .. \otimes P_2$$

the combination sequence order does not matter, so if an optimal covering hypertree can be found then the potential can be combined. As research [36] on optimal covering hypertree concluded that finding this optimal combining sequence is an NP[1] hard problem, heuristic operations for hypertree manipulation has been proposed.

3.5.1 Local computation and the joint tree construction

[1] nondeterministic polynomial time

Valuation networks [Shenoy 1989] is based on an algebraic structure for local computation called valuation algebra which is a formalism of solving inference problems based on graphical structure called joint tree. It has been shown that Dempster Shafer theory perfectly fits into the valuation framework [36] and has been successfully been utilized for reducing the number of focal elements in the state space with eliminating variables in the network, so the Dempster's rule of combination becomes feasible even with complex and large domains where the number of variables exceeds 32 or more.

However finding an optimal elimination sequence also proved to be a NP hard problem[1] so several algorithms has been examined to overcome the above mentioned difficulties.

In valuation network the knowledge is represented by valuations. This knowledge can be a probability assignment from Bayes theory or belief or mass functions from the Dempster Shafer or possibility theory. Inference in the network can be carried out using two basic operations as follows:

- Combination-aggregation of the knowledge
- Marginalisations-coarsening of the knowledge

The components of the valuation framework:

Variables and configuration

Lets consider a finite set of variables $x_1 \ldots x_n$ on the frame of D where $x_n \in D$. All variables has a finite set of values denoted by Θ_x

Valuations

A valuation ϕ represents knowledge about the variables in D and $d(\phi) = D$ represents the domain of ϕ. Concerning the Dempster-Shafer theory the valuation ϕ can correspond to mass function or belief function.

[1] Solvable in polynomial time by a nondeterministic Turing machine

Combination

Combination is a function $(\phi,\varphi)\mapsto\phi\otimes\varphi$ where ϕ and φ are valuations for D_1 and D_2 and $\phi\otimes\varphi$ is a valuation for $D_1\cup D_2$

Marginalisation

Marginalisation is a function $(\phi,D')\mapsto\phi^{\downarrow D'}$ where $D'\subseteq d(\phi)$ and ϕ is a valuation for D where $D'\subseteq D$ therefore $d(\phi^{\downarrow D'})=D'$

The idea behind the local computation is to limit the particular operations into smaller domains with variable elimination so the computation can be carried out locally which makes complex computational scenarios feasible. Variable elimination results in a graphical structure called joint trees which consists of a set of nodes with the initial valuations distributed on them where nodes are connects to each other and computations are carried out by message passing between the nodes. This computational scheme has been realized and investigated with different architectures.

- Shenoy-Shafer[37] architecture
- Lauritzen-Spiegelhalter[38] architecture
- Hugin[39] architecture
- Fast Division architecture [40]

A detailed comparison of these architectures can be found in [36].
Different elimination sequences lead to different joint tree structures and as it was mentioned before determining the optimal elimination sequence is a NP hard problem different heuristics has been developed that address the problem. Further constraint is that joint trees has to maintain the so called Markov property that requires:
$D_i\cap D_j\subseteq D_k$ for every pair of nodes $N_i;N_j\rightarrow N_k\in Path(N_i,N_j)$.

Suppose we have valuations figure 6 on the following domains: {a, b}, {b, c}, {c, d}.

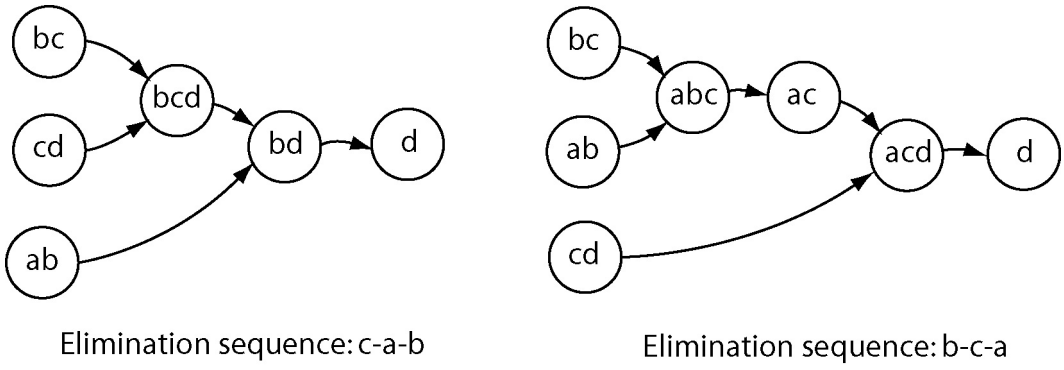


Figure 6.

The following hypergraph[1]-hypertree[2] based heuristics has been investigated:

- OSLA-Smallest Clique: if possible eliminates a leaf in every step.
- OSLA –Fewest Fill Ins: defines a fill in number $FI(x,\gamma)$ for the variable x to be eliminated from γ with a number of pairs $x_i, x_j \in N(x,\gamma)$, which are not connected. The algorithm eliminates variables first where the fill in number is as small as possible.
- OSLA- Smallest Clique- Fewest focal sets[3]: combination of the above mentioned with defining the smallest focal set instead of the fill in number.

I believe that given the generic nature of valuation algebra that makes local computation possible is the prosperous alternative for addressing the usability problem on large and complex domains of the Dempster ‘s rule of combination. However research has to be done in order to evaluate the applicability of the framework in a distributed environment. Further needs to be proved that extending valuation algebra into the distributed domain, will keep the important properties intact and can improve the performance and the effectiveness of such

[1] A graph in which generalized edges may connect more than two nodes
[2] An acyclic hypergraph
[3] Belief functions which value is not equal to 0

systems since the problem of ontology mapping in a multi agent environment arises in many scenarios.
From the technical point of view I expect improvements in both time and accuracy of the mapping process so that question answering with using different ontologies can provide real time mapping and response to the user.

3.6 Analysis of previous work

Ontology mapping is a widely investigated area and numerous approaches led to different solutions. To date uncertainty handling during the mapping process was not in the focus of the research community since initially only different logic (FOL,DL) based approaches has been utilized. As practical application of ontologies emerged on the web it has been acknowledged that considering the dynamic nature of the Web the problem of inconsistencies, controversies and lack of information needs to be handled. First systems that used probabilistic information like LSD, GLUE proved that combining different similarity measures based on probability could significantly improve the accuracy of the mapping process. I believe that probability theory and distribution does not have enough expressive power to tackle certain aspects of the uncertainty e.g. total ignorance. As a consequence I think evidence (Dempster-Shafer) theory is the most suitable approach and needs to be investigated in ontology mapping context though this has not been done so far. The reason is that Dempster Shafer combination rule can easily be unfeasible in case of domains with large number of variables. Different optimalisations methods have been developed but to date I could not find approaches that considered distributed environments. Local computation and valuation networks uses joint tree structure to narrow down the number of focal sets (elements) and different architectures have been proposed based on message passing schemes to carry our inference and resolve the problem of the Dempser's rule of combination. In my scenario I assume a dynamic multi agent environment where different agents have partial knowledge of the

domain. I believe that valuation network is a prosperous candidate for my scenario however the problem of distributed knowledge and inference with its implications e.g. distributed joint tree construction needs to be addressed in my Ph.D. research.

4. Pilot project work

4.1 Context

European Commission-Joint Research Center Institute of Energy, Petten provides scientific and technical support for the conception, development, implementation and monitoring of community policies related to energy. One of the main competencies of the Nuclear Safety Unit (NSU) is to ensure the structural integrity of components for safe operation of nuclear facilities with focus on specific issues such as steel components, inspection qualification, ageing of materials. Several databases (AlloysDB, GasketDB, CorrosionDB, HTR-FUELDB) store mechanical and physical properties of engineering materials produced by the European RTD projects that covers the materials behavior at low, elevated and high temperatures for base materials and welded joints and also includes irradiation materials testing in the field of fusion and fission and thermal barrier coatings tests for gas turbines. The above-mentioned databases will be the data sources of my prototype. The context of the research is the AQUA query answering system where a user poses a query against the Meta descriptor and the plan generator based on the source description rewrites the query that is executed by the execution engine.

The purpose of my research is to extend the AQUA system with a multi agent environment that make use of the heterogeneous information sources described by their own ontology.

On the user interaction level the user poses a natural language query against a Meta descriptor that is represented by FOL (First Order Logic) formulas by the AQUA system. To specify a query AQUA uses ontologies and the WorldNet database in order to clarify the questions during the interactive dialog with the user. Once the query has been built up FOL formulas are created and passed onto the mediator system.

On the mediator level the query reformulation engine receives the FOL formulas and reformulates it into the Meta descriptor representation. The Meta descriptor describes what information can be found in the different ontologies. The Meta descriptor is neither the union of the local ontologies nor reference ontology, only a description of what kind of information is in the local ontologies. This can be used to direct the query to the relevant source.

Based on the reformulated query, a broker agent decomposes the query into sub-queries that will be passed into the mapping agents. Broker agents are natural query agents that can handle one kind of query (e.g. what, where, who). Mapping agents through the resource agents obtain the relevant information that can answer the query. Mapping agents need to map the concepts and attributes between the different local sources that correspond to the query terms that are based on the Meta descriptor.

My research focuses on a scenario when different sources contain the whole or partial information to answer the query. In this context two main interesting research areas can be identified namely query decomposition and ontology mapping. While query decomposition itself constitutes a separate research, establishing ontology concept mapping with layered multi agent architecture formulates the core of the Ph.D. research where the semantic mapping is being built up dynamically in run time through communication acts between the neighboring layers namely the brokering and the source layer.

To explain the problem we need to consider the differences in context between the ontology mapping in the database and the WWW domain.

In case of databases the data is described by schema that represents the content of the resource on the structural level. Ontologies on the WWW go one step further because it describes information on the semantic level that is the meaning of terms used in the domain. Usually there is a consistent and direct connection between the database schema naming structure and the data itself but this is not always the case. One can realize differences in the use of matching for

Databases:

- Information is represented by concrete data records that are embedded into the document
- Data centric retrieval methodologies

Semantic web-ontology:

- Information is scattered across the document and are present in a form of text
- Document centric retrieval methodologies.

First phase ontology research were successfully proved that the use of metadata and ontology in the Web document can improve the search capabilities of a particular search engine so the user can retrieve more relevant documents of his/her interest [56]. Ontology mapping in this case is used to detect similar concepts or synonyms in the document structure and the query so a more intelligent search could be achieved. I believe that we need go further than this especially when we consider a query answering system that uses information from on-line databases. The reason is that in the latter scenario it is not enough to make a rough mapping between similar concept or properties between the database fields and the query string but we need a reliable and dynamic semantic mapping in order to give accurate answer for the users query. As an example consider if one looks for thermo hydraulic data from a specific experiment. If all experiments are returned which has temperature component stems from the thermo keyword in the query then our system will be easily overwhelm the user with a vast number of irrelevant information.

4.2 Ontologies

Ontologies that describe the entities in the different databases are under development and cover the main concepts like test result, source, material,

specimen, test condition, etc. Ontologies were created using the Protégé tool and are present in OWL format.

Now suppose that in a database different agents are responsible for a specific area/concept e.g. agentDB1-Ontology1 1-materialDB1-Ontology1, agent DB1-Ontology1 2-specimenDB1-Ontology1. and agent DB2-Ontology2 1 material DB1-Ontology1,agent DB2-Ontology22-Test DB2-Ontology2. Agent 1 and agent 2 know that according to their own ontology there is a relation between material and test but the representation of the relation can be different in the different ontologies.

4.3 Queries

On a high level the classes describing the ontology are source, material, specimen, test condition and the test. All these entities are described by subclasses that the specific query can be related to.
The scope of the queries can be classified by the information that the query needs to answer:

- Which (source, material, specimen, test condition, test): This type of query can be answered by the identifier of a particular entity and it corresponds to the existence of it e.g. Which material has been tested under stress controlled thermo mechanical fatigue?
- What is or List the value (range) of the (stress, strain, temperature): This type of queries can be answered by a value or values by a specific property of the entity and it relates to a specific property or properties of the entity e.g. What is the stress range of the CR12 MO material under uniaxial creep?

4.4 Formalizing queries in FOL

Broker agent receives a query represented in FOL from interface agent that can be divided into sub queries such as follows:

- Which tests has been carried out on a bar shaped specimen?
  (∀x, ∃y)Test(x) and Specimen(y) and form(y,bar) and carriedOutOn(x,y)
- What kind of tests has been carried out on material that has name 10 CrMo 9 10?
  (∃x,∀y)Test(x) and Material(y) and name(y, 10 CrMo 9 10) and carriedOutOn(x,y)
- List all materials that have been tested under in hydrogen test environment?
  (∀x, ∃y)Material(x) and Test(y) and hasEnvironment(y,hydrogen) and hasMaterial(y,x)
- What are the elastic strain results for material 10 CrMo 9 10 under thermo mechanical fatigue.
  (∃x, ∃y) Test(x) and Material(y) and testType(x,Thermo-mechanical fatigue) hasName(y, 10 CrMo 9 10) and and hasCurve(x,elastic-strain) and hasMaterial(x,y)
- List materials with test control strain resistant heating?
  (∀x, ∃y) Material(x) and Test(y) and hasTestControl(y, strain resistant heating) and hasMaterial(y,x)
- Which materials have bar shaped specimen that has been tested for uniaxial creep?
  (∀x, ∃y, ∃z)Material(x) and Specimen(y) and Test(z) and hasShape(y,bar) and hasTestType(z,uniaxial creep) and hasMaterial(z,x) and hasSpecimen(z,y)
- Which cylindrical shaped specimens have strain induction heating control under thermo mechanical fatigue?

(∀x, ∃y)Specimen(x) and Test(y) and hasShape(x,cylindrical) and hasTestType(y,Thermo-mechanical fatigue) and hasSpecimen(y,x)

- Which materials have been tested in hydrogen environment for cyclic creep?

  (∀x, ∃y)Material(x) and Test(y) and hasEnvironment(y,hydrogen) and hasTestType(y,cyclic creep) and hasMaterial(y,x)
- What are the maximum strain results under low cycle fatigue for Fe-base alloys?

  (∀x, ∃y)Test(x) and Material(y) and hasTestType(x,low cycle fatigue) and and hasCurve(x,maximum strain) and hasClass(y,Fe-base alloy) and hasMaterial(x,y)
- Which project partners tested 10 CrMo 9 10 material for multiaxial creep on a thin walled tube specimen?

  (∀x, ∃y, ∃z, ∃v) Source(x) and Material(y) and Specimen(z) and Test(v) and hasName(y, 10 CrMo 9 10) and hasTestType(v,multiaxial creep) and hasName(z,thin wall tube) and hasSource(v,x) and hasMaterial(v,y) and hasSpecimen(v,z)
- What are the inelastic strain results for material that contains Cr from JRC Petten source under high cycle fatigue?

  (∀x, ∃y, ∃z) Test(x) and Material(y) and Source(z) and hasTestType(x,high cycle fatigue) and hasCurve(x,inelastic strain) and hasName(z,JRC Petten) and hasElement(y,Cr) and hasSource(x,z) and hasMaterial(x,y)
- Which plate form specimens have low cycle fatigue test with DIM 332 standard?

  (∀x, ∃y)Specimen(x) and Test(y) and hasStandard(y,DIM 332) and hasTestType(y,low cycle fatigue) and hasSpecimen(y,x)
- Which high cycle fatigue tests produced transgranular fracture for materials where the producer was Alstom?

(∀x, ∃y)Test(x) and Material(y) and hasTestType(x,high cycle fatigue) and hasFracture(x,transgranular) and hasManufacturer(y,Alstom) and hasMaterial(x,y)

- Which materials with directionally solidified forming process were tested for any kind of creep on Instron III test machine?
  (∀x, ∃y) Test(x) and Material(y) and hasFormingProcedure(y,solidified) and hasTestMachine(y,Instron III) and hasTestType(x,Creep) and hasMaterial(x,y)
- Which test results produced brittle fracture for materials that went through homogenizing thermo mechanical heat treatment
  (∀x, ∃y) Test(x) and Material(y) and hasHeatTreatment(y,homogenizing) and hasFractureType(x,brittle) hasMaterial(x,y)

4.5 Remarks

The main problem during the mapping process is that not all relation and concept names appear in the knowledge base. Therefore there is a need for similarity algorithms. Section 4.6 describes a preliminary string similarity algorithm used in the initial prototype.

4.6 Similarity

4.6.1 Syntactic similarity

To assess syntactic similarity between ontology entities we use different string-based techniques to match names and name descriptions. These distance functions map a pair of strings to a real number, which indicates a qualitative similarity between the strings. To achieve more reliable assessment we combine different string matching techniques such as edit distance like functions e.g.

Monger-Elkan[58] to the token-based distance functions e.g. Jaccard[59] similarity.

To combine different similarity measures we use Dempster's rule of combination (see section 4). There are several reasonable similarity measures exist, each being appropriate to certain situations. To maximize our system's accuracy we employ a broad variety of similarity measures. At this stage of the similarity mapping our algorithm takes one entity from Ontology 1 and tries to find similar entity in extended query. The similarity mapping process is carried out on the following entities:

- Concept-name similarity
- Property set similarity

The use of string distances described here is the first step in identifying matching entities between query and the ontology or between ontologies with little prior knowledge, or ill structured data. However, string similarity alone is not sufficient to capture the subtle differences between classes with similar names but different meanings. So we work with WordNet in order to exploit synonymy at the lexical-level. Once our query sting is extended with lexically synonym entities we calculate the string similarity measures between the query and the ontologies.
In order to increase the correctness of our similarity measures the obtained similarity coefficients need to be combined. Establishing this combination method is the primary objective that needs to be delivered with the pilot study. Further once the combined similarity has been calculated we need to develop a methodology to derive a belief mass function that is the fundamental property of Demster-Shafer evidence theory.

4.6.2 Semantic similarity

In our prototype it is necessary to assess not only the syntactic but also the semantic similarity between concept, relations and the properties.

For semantic similarity between concept, relations and the properties we use graph-based techniques. We take the extended query and the ontology input as labeled graphs. The semantic matching is viewed as graph-like structures containing terms and their inter-relationships. The similarity comparison between a pair of nodes from two ontologies is based on the analysis of their positions within the graphs. Our assumption is that if two nodes from two ontologies are similar, their neighbors might also be somehow similar. We consider semantic similarity between nodes of the graphs based on similarity of leaf nodes. That is, two non-leaf schema elements are semantically similar if their leaf sets are highly similar, even if their immediate children are not.

The main reason why semantic heterogeneity occurs in the different ontology structures is the fact that different institutions developed their data sets individually, which contains mainly overlapping concepts. Assessing the above-mentioned similarities in our multi agent framework we adapted and extended the SimilarityBase and SimilarityTop algorithms [43,44] used in the current AQUA system for multiply ontologies. The goal of our approach is that the specialized agents simulate the way in which a human designer would describe its own domain based on a well-established dictionary. What also needs to be considered when the two graph structures obtained from both the user query fragment and the representation of the subset of the source ontology is that there can be a generalization or specialization of a specific concepts present in the graph which was obtained from the local source and this needs to be handled correctly. In our multi agent framework the extended and combined SimilarityBase and SimilarityTop algorithms can be described as follows:

1. Using WordNet an extended directed graph is constructed from the FOL query fragment where there are bi-directional edges between the nodes representing the concepts and there are directed edges from the concepts to the property nodes. In this step the specialized agents try determine all possible meanings of the query fragment. Figure 1 depicts the graph

representation of the hasName(material, 10 CrMo 9 10) FOL query fragment.

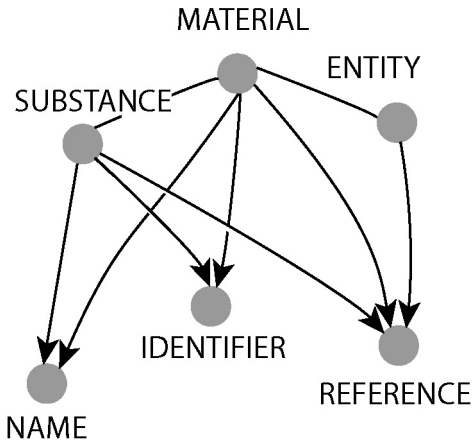


Figure 7. G_0 query fragment graph

1. Based on different string similarity measures (see section 3.1) the specialized agent builds up a directed graph from the local ontology structures that supposedly answer the query fragment. Figure 2 depicts two graph fragments with belief functions obtained from two different ontologies where G1 is the graph located to left hand side of the middle bar and G2 is located on the right hand side.

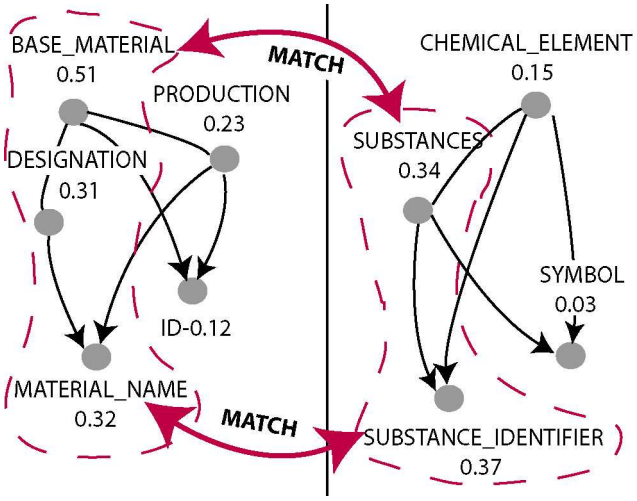


Figure 8. G1 and G2 graph representation of the local ontology fragment

2. Top-down sub-graph (isomorphism) similarity assessment is applied on the graph G_0 in order to find the subgraph G_1 and G_2 respectively. The aim is to find identical subgraphs to G_1 and G_2 in order to assess the similarity of the concepts and properties that can answer the query fragment. We call this method a top-down assessment because the search for the sub graphs starts from the concept nodes towards property nodes through the directed edges. Once we reached the property node the search stops. If along the path we walked through the graph we found a sub graph identical (isomorph) to G_1 and G_2 that agent can deduce that the query fragment can be answered from the sources that belong to the particular ontology and the concepts or properties identified in the different sources are similar to both each other and to the query fragment and a basic mass function can be calculated that express the extent of belief in the existence of the similarity mapping between them. In case G_1 or G_2 contains nodes that could not be found in the G_0, because of the nature of the top down assessment the agent can deduce that the particular concept node is a specialization of the concept that was identified by the algorithm.

In order to achieve the best possible similarity measures we carried out an experiment to determine which similarity algorithm would provide the best measure in our scenario. We examined 10 different algorithms on 800 entities (concepts and properties) and carried out numerous tests with different ontologies. As an example the following concept names are included in the before mentioned test:

- Fracture-time and Time-at-Fracture
- Test-Standard and Standard
- Hold-Time-At-Maximum-Temperature and Maximum-Temperature At-Hold-Time
- Axial-Engineering-Strain and Strain-Axial-Eng

The tested similarity algorithms were as follows:

1. SLIMWinkler is the combination of the Same Letter Index Mixture and the Winkler algorithm.
2. CharJaccard

$$sim_{x_a x_b} = \frac{x_a * x_b}{\|x_a\| \|x_b\| - x_a * x_b}$$

where $x_a * x_b$ is the inner product of x_a, x_b and $\|x\|$ is the Euclidean norm for the vectors.

3. Jaro

$$jaro_{x_a x_b} = \frac{1}{3}\left(\frac{|x_a^{'}|}{|x_a|} + \frac{|x_b^{'}|}{|x_b|} + \frac{|x_a^{'}| - T_{x_a^{'}, x_b^{'}}}{|x_a^{'}|}\right)$$

where $x_a^{'}$ are the characters in x_a which are common with x_b and $T_{x_a^{'}, x_b^{'}}$ is half the number of transpositions for $x_a^{'}, x_b^{'}$.

4. Level2JaroWinkler is the recursive version of the JaroWinkler
5. Level2MongleElkan is the recursive version of MongleElkan

$$Monge - Elkan(S,T) = \frac{1}{|S|}\sum_{i=1}^{|S|}\max_{j=1}^{|T|} monge - elkan(S_i, T_j)$$

6. Level2SlimWinkler is recursive version of SlimWinkler
7. SLIM is same-letter index mixture distance where
8. JaroWinkler

$jaro - winkler_{x_{a,x_b}} = jaro_{x_{a,x_b}} + \frac{P^{'}}{10} \cdot (1 - jaro_{x_{a,x_b}})$ where P is the length of the longest common prefix of x_{a,x_b}

9. UnsmootherJS is Jensen-Shannon distance of two unsmoothed language models and

$$jensen - shannon_{x_a x_b} = \frac{1}{2}(KL(P_{x_a} \| Q) + KL(P_b \| Q))$$

where $KL(P_{x_a} \| Q)$ is the Kullback-Lieber divergence

10. TFIDF

$$TFIDF_{x_a x_b} = \sum_{\omega \in x_a \cap x_b} V(\omega, x_a) \cdot V(\omega, x_b)$$

where TF_{ω, x_a} is the frequency of word ω in x_a, x_b and IDF_ω is the inverse of the fraction of names in the corpus that contain ω and

$$V(\omega, x_a) = \frac{\log(TF_{\omega, x_a} + 1) \cdot \log(IDF_\omega)}{\sqrt{\sum_{\omega'} (\log(TF_{\omega, x_a} + 1) \cdot \log(IDF_\omega))^2}}$$

We run 3 experiments in order to assess the correctness of the before mentioned algorithms. We generated 3 different versions of the 800 entity names and we divided it into 3 sets. We carried out similarity assessment with the before mentioned algorithms and calculated an average of the 3 different similarity measures.

- Test 1: We permutated the terms in the entity name order or used the plural form of the same entity. This provides us a very realistic scenario to which we can easily face in our proposed system e.g. Fatigue-Crack-growth and Crack-Growth-Fatigue or Source and Sources
- Test 2: We removed certain characters from the entity names e.g. Fatigue-crack-growth and CrackGrowthFatigue or Sources and Source.
- Test 3: We introduced random errors into the entity names e.g. Fatigue-crack-growth and crackOgrowthOFatigue or Source and SourceQ

Similarity measures were assessed to all concepts and properties that were incorporated into the query and ontology graphs. A visualization of these two graphs is depicted on figure 9. On the left window the Query graph is represented. It contains all the additional concepts and properties that have similar meaning comparing to the query terms. On the right window the ontology graph is represented. It contains all concepts and properties from an ontology that is similar to any of the entities in the query graph.

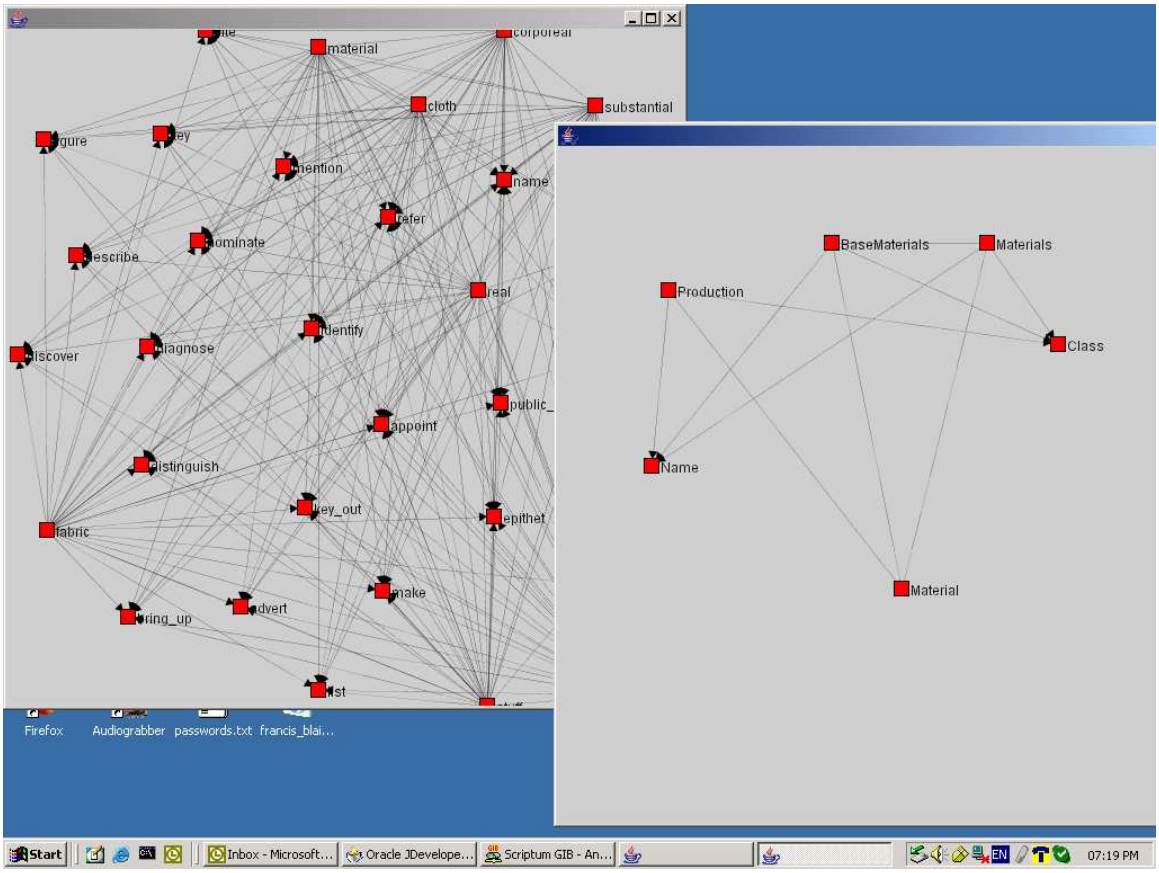


Figure 9. Query and ontology graph corresponding to a particular query.

The experiment gave us a quantitative comparison of how these algorithms will perform in a realistic scenario. Our aim is to select the 3 best performing method and incorporate it into our proposed system.

The result of 3 test series and the average of the different tests are depicted on figure 10.

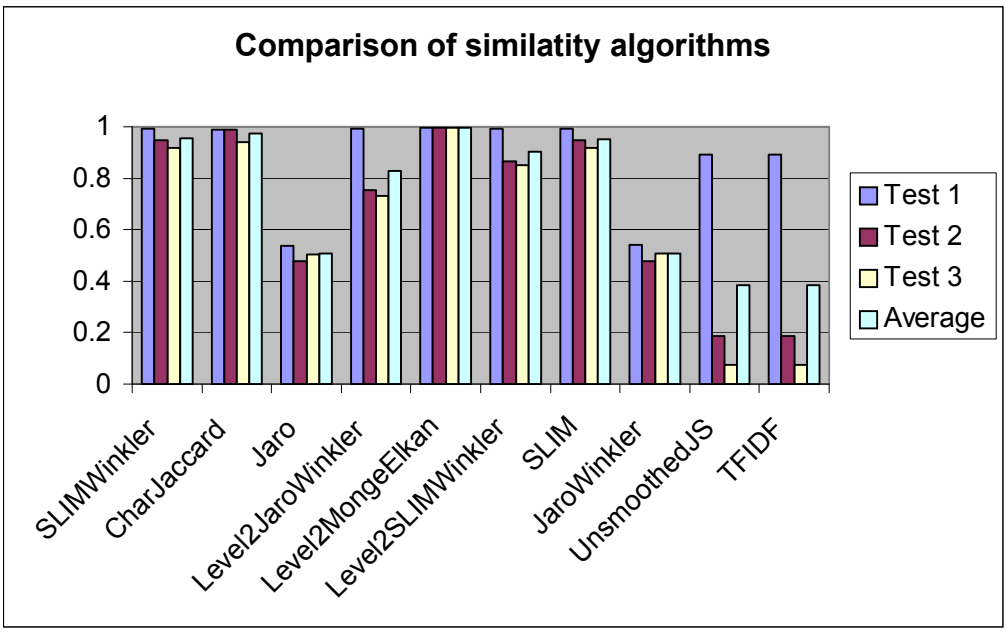


Figure 10. Comparison of similarity algorithms

We considered two ontologies from the same domain. Appendicies shows two different ontology fragments. Both ontology contains entities like test and specimen however both the naming conventions and node hierarchy are different

4.6.3 Combining similarity measures with Dempster's combination rule

An important aspect of the mapping is how one can make a decision over how different similarity measures can be combined and which nodes should be retained as best possible candidates for the match. To combine the qualitative similarity measures that have been converted into belief mass functions we use the Dempster's rule of combination and we retain the node which belief function has the highest value. Our algorithm takes all the concepts and its properties from the different ontologies and assesses similarity with all the concepts and properties in the query graph. Imagine the scenario mentioned in section 3.2. In the ontology graph G1 we take the node "BASE_MATERIAL" and utilizing different similarity measures (see section 3.1) we receive two similarity graphs

where figure 11 depicts the CharJaccard and figure 12 shows the Monger-Elkan similarity measures that has been obtained by comparing the "BASE_MATERIAL" node to concepts in our extended query graph (like "MATERIAL, SUBSTANCE, ENTITY"). CharJaccard Similarity uses letter sets from the comparison instances to evaluate similarity and the Monge Elkan approach makes other additional tests by taking the semantic similarity of a number of fields and sub fields into consideration.

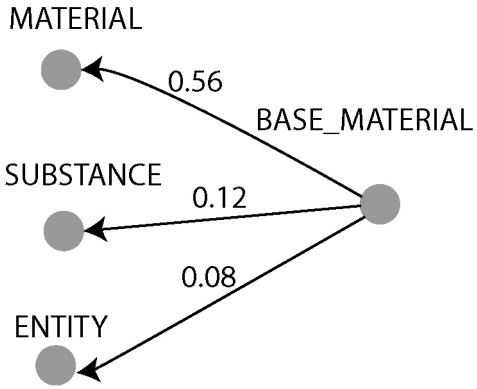


Figure 11. Obtained similarities based on CharJaccard similarity

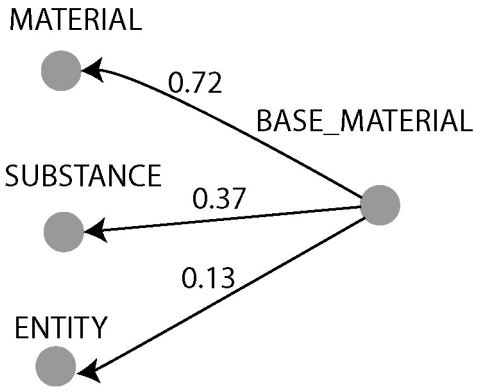


Figure 12. Obtained similarities based on Monger-Elkan similarity.

To obtain more reliable results we need to combine the similarity assessments based on the different similarity measures (figure 11,12). The combination of the different similarity measures is not a straightforward question and it includes several biases. Our approach is to consider these measures as subjective probabilities and utilize a well-established framework that provides convenient way to represent and combine our evidences.

The Dempster's rule of combination provides a well-established approach however it works with belief mass functions where the sums of the masses add up to 1. To convert similarities into belief masses we need to normalize the problem space into 1. This way we can easily obtain the necessary masses so we can utilize the combination rule.
Instead of just retaining nodes where the belief mass function exceeds a certain predefined limit we can examine leaf nodes from the graph and calculate a belief that will give us a good indication which node needs to be retained as the best possible matching candidate.
Using the above-mentioned process we generated the combination of the similarity measures in the graphs (figure 11-12) with the necessary belief mass functions which can be used to calculate the belief in a particular proposition e.g. the highest belief shows that MATERIAL and BASE_MATERIAL is a definite match.

Seeking to achieve the best possible mappings we tested how the conversion of similarity measures into belief functions and its combination would affect the matching accuracy in our scenario. We applied 3 different similarity measures (Char Jaccard, SLIM Winkler and Level2 Monge-Elcan) on 800 entities (concepts and properties) and carried out numerous tests with different sets of properties and concepts. We run 3 experiments in order to determine how belief and combination of belief can affect the original similarity measures. We generated 3 different versions of the 800 entity names and we divided it into 3 sets. Than we choose a changed entity which represents a query fragment and selected the original ontology concept plus 8 similar concepts from the ontology (represents the extended query). Finally we applied different similarity measures and calculated the belief functions on these entities. Our objective was to determine the accuracy of the mappings. The experiment gave us a quantitative comparison of how the belief assessment would perform in a realistic scenario. Our long-term objective is to identify which factors affect the uncertainty and how can we improve it in the context of ontology mapping

The result of 3 test series and the combined belief are depicted in figure 13.

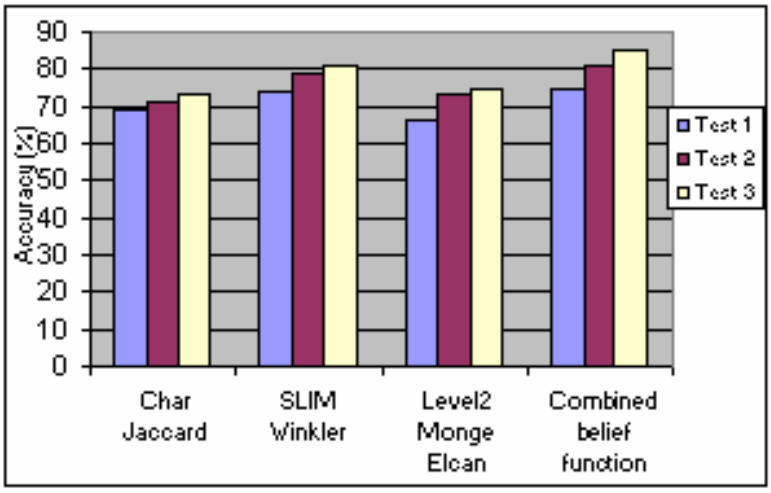


Figure 13. Combined belief and similarity measure comparison

4.7 Uncertainty handling algorithms

An important part of the system is how the similarity measures are applied in the concrete scenario and how the particular agent assesses the belief mass functions and belief functions. In our experimental system we consider basic probability assessment over the following entities:
Class: The most basic concepts in the domain that correspond to classes that are the root of the various taxonomies
Object properties: Relation between the instances of two classes
Data type properties: Relation between instances of classes and RDF literals and XML Schema data types therefore it describes that the particular class e.g. material has a data type property called name which is a string.

To describe the algorithms we define the followings:

Definition 1: The query fragment is $\tau = (c, o, d)$ where c is the concept present in the partial query posed by the user, o is the object property and d the data type property.

Definition 2: Given 2 source ontologies $o = (C, O, D)$ and $o' = (C', O', D')$ where C is the set of concepts, O is the set of object properties and D is the set of data type properties in the ontologies respectively.

Definition 3: The assessed belief mass functions are $\beta_1 = (c, \{C, C'\})$, $\beta_2 = (o, \{O, O'\})$, $\beta_3 = (d, \{D, D'\})$ for the set of concepts (β_1), object properties (β_2) and data type properties (β_3) represented in the source ontologies.

Figure 14 describes the main uncertainty-handling algorithm carried out during the mapping process:

Input: $\tau = (c, o, d)$
/ Initial hypothesis assessment */*
C,C'=CONCEPTSSELECTION
O,O'=OBJECTPROPERTYSELECTION
D,D'=DATAPROPERTYSELECTION
1:while not found do
    2: similarity assessments c, C
    3: similarity assessments c,C'
    4: similarity assessments C,C'
5: if found then
    6: similarity assessments o, O
    7: similarity assessments o, O'
    8: similarity assessments O, O
    9: similarity assessments d, D
    10: similarity assessments d, D'
    11: similarity assessments D, D'
12:end while
13:combine basic belief functions with other agents
14: determine probable answers for the query
15: assess belief for the probable answers

Figure 14.

4.8 System architecture

The high-level system architecture (figure 15) shows how the functional parts of the system are related with each other. In the mediator layer the agents are organized in different levels. Agents on the broker level responsible for decomposing the query into sub queries. The decomposed query parts are sent into the mapping agents located in the mapping layer. Mapping agents obtain the relevant information from the sources through the source agents. When only one source corresponds to the query the scenario is pretty straightforward and there is no need for any mapping between the sources, the query can be answered from the source. In a real case scenario this possibility is not so likely and this is why the mapping between local ontologies is a justified scenario in our case.

Concerning the environment the JADE agent development framework is the best fit for the research need. Additional advantage that KMi has experience in JADE framework gained from earlier research projects.

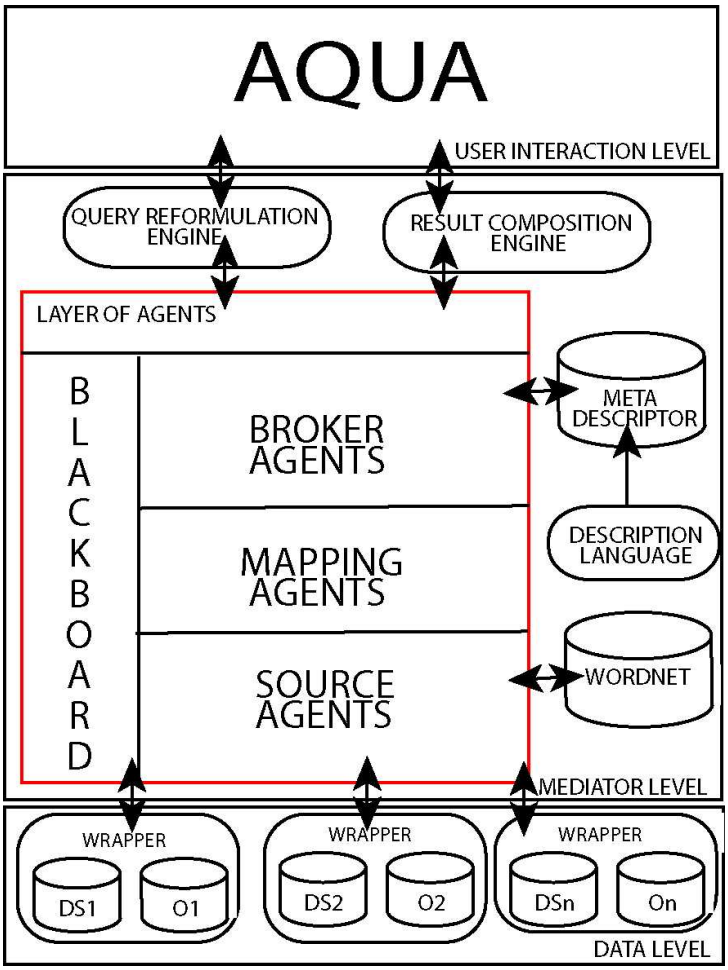


Figure 15.

Since the research will focus on the mapping part of the problem detailed agent architecture is depicted in figure 15. My Ph.D. research will be confined to the part that is framed with red color on the general framework architecture. The idea that will be investigated in my research is the mapping agents can build up mappings simultaneously, utilizing different similarity measures. Based on their belief agents need to harmonize their beliefs based on trust that is formed during the mapping process.

This is a two-step process:

1. Mapping agent based on evidences that are available to them built up belief about the mapping.
2. Group of mapping agents need to harmonize their beliefs over the solution space.

The key components of the prototype are grouped by the different functional levels and from bottom to up as follows.

Data Level

On the data level the heterogeneous data sources are represented by their ontologies. The format of these sources varies from relational databases to simple files.

- Data source (DS): actual data represented in the database, file etc.
- Ontology (O) Semantic metadata that describes the particular data source.
- Wrapper creates a unified XML representation of the source that is queried by the particular resource agents.

Mediator level

- Layer of agents (figure 16): Typically three kind of agents:
    - **broker** that receives a FOL query and decomposes it into sub queries.
    - **mapping** that has knowledge of a particular domain specific area and cooperatively map up source concept with the concepts contained by the query string.
    - **source** that access a particular data source and it's ontology and passes it to the mapping agents on a request basis.

  The first JADE implementation is shown in Appendicies. The snapshot depicts both the available agents and the communication between them in our prototype system.
- Meta descriptor and description language: Key component of the system that describes what kind of information can be found in the different

sources. Practically FOL knowledge base that contains information about relations of the local resources. As an example let's consider a query where one is looking for materials with a specific name. The meta descriptor will contain information that e.g. Ontology1 and Ontology2 describes Material related entities.

- Blackboard: A blackboard is a task independent architecture for integrating multiple knowledge sources e.g. different local agents. Task independent means that it can be used for a wide range of tasks. In a blackboard system, a set of knowledge sources share a common global database (blackboard). The contents of the blackboard are often called hypotheses. Knowledge sources respond to changes on the blackboard, and interrogate and subsequently directly modify the blackboard. This modification results in the creation, modification and solution of hypotheses. Because of only knowledge sources are allowed to make changes to the blackboard it is through the blackboard how the knowledge sources communicate and cooperate. The blackboard holds the state of the problem solution, while the knowledge sources make modifications to the blackboard when appropriate.
- Query reformulation and result composition engine: A query that is raised by the user needs to be reformulated and decomposed before entered into the system, which is the purpose of the query reformulation engine. Information flow stems from the mapping process needs to be composed into a single coherent answer, which is done by result composition engine. These subsystems are out of the scope of my research.

User Interaction level

The AQUA[45,46] query answering system itself, which provides precise answers to specific questions raised by the user. It integrates Natural Language Processing (NLP), Logic, Ontologies and Information retrieval techniques. I believe that in the AQUA question-answering context that exploits Semantic Web technologies a distributed multi agent ontology mapping can prove the practical

applicability of the Dempster-Shafer theory of evidence for complex domains with large number of variables. In our prototype AQUA is the interface that provides First Order logic predicates to our broker agent based on the users query.

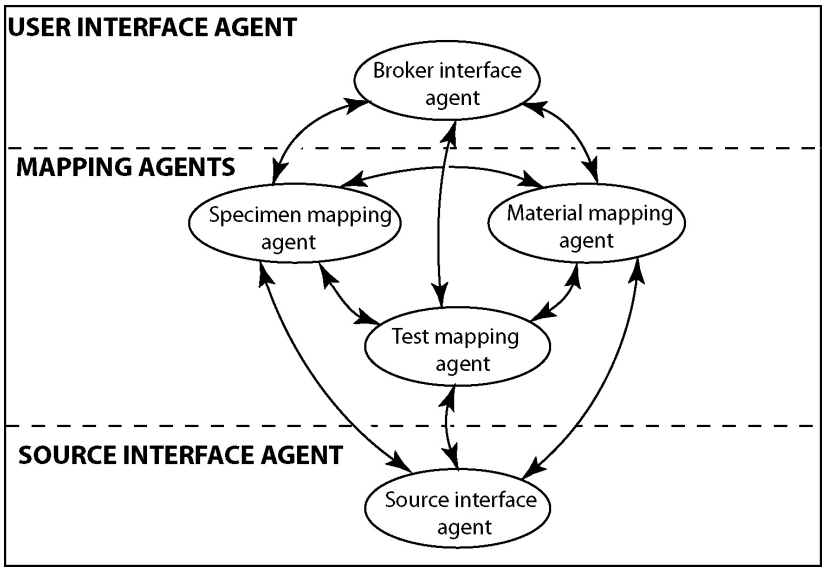


Figure 16.

4.9 Mappings, input and output for the working example

In the following chapter a detailed example is presented how our mapping framework carries out its functional operations from the broker agent to the mapping. We consider the following query as an **input** to our mapping framework:

“Which test has been carried out on a bar shaped specimen?” with its FOL representation:

$(\forall x, \exists y)$ (Test(x) and Specimen(y) and form(y,bar) and carriedOutOn(x,y))

The **mappings** that will be identified in two different ontologies by the particular agents are:

1. Test agent:

ONTOLOGY 1	ONTOLOGY 2
Test	TestResult
Control	TestControl
Temperature	TestTemperature
Standard	TestStandard

2. Specimen agent

ONTOLOGY 1	ONTOLOGY 2
Specimen	Specimen
Form	SpecimenForm
Name	SpecimenName
Characterisation	SpecimenCharactesisation

Outputs will be the concrete class or property instances in the ontology that has been identified by the mappings.

4.10 Working example

The Meta descriptor describes what kind of information can be found in the different local ontologies/sources.

$$MD = DO_1 \bigcup DO_n \bigcup DO_{n+1}$$

where MD is the Meta descriptor and DO_n is one of the particular domain ontology and

$$DO_i = \{R_{i1} ... R_{ij}\}$$

where R_{ij} means the relation j in the ontology i

As discussed the Meta descriptor can be best represented by FOL since the AQUA system also creates the query in FOL.

The Blackboard contains information about:

- Agents (MaterialAgent, SpecimenAgent, etc.) as constant symbols
- Query and property information (canAnswer(x,Test), hasInformation(x,MaximumStress)) as predicate symbols.

To better illustrate the problems that the research and the prototype system will address the following events are detailed:

1. At system startup the knowledge of the broker contains only the pre-defined concept-mapping agent pairs that describe which agent knows the particular concept:

   ∀x MaterialAgent(x) and canAnswer(x,Material)

   ∀x SpecimenAgent(x) and canAnswer(x,Specimen)

   ∀x TestAgent(x) and canAnswer(x,Test)

   ∀x SourceAgent(x) and canAnswer(x,Source)

   ∀x TestConditionAgent(x) and canAnswer(x, TestCondition)

2. FOL Query passed to the broker agent:

   Which test has been carried out on a bar shaped specimen?

$(\forall x, \exists y)$ (Test(x) and Specimen(y) and form(y,bar) and carriedOutOn(x,y))

3. Broker agent decomposes the query into sub queries and forwards it to the particular agents:
   - TestAgent receives Test(x) and carriedOutOn(x,y)
   - SpecimenAgent receives Specimen(y) and form(y,bar) and carriedOutOn(x,y)

   Both agent received part of the query that corresponds to multiple entities. Since this is a relation between the two concepts, agents need to share the meaning of this expression. Agents place this into a blackboard, which is visible for all agents.
   - carriedOutOn(x,y) added to the Blackboard
4. Test and Specimen agents retrieve fragments of two ontologies. Test Agent identifies two similar concepts:
   - O1 contains TestResult and O2 contains Test

   Specimen Agent identifies two similar properties:
   - O1 contains Form and O2 contains SpecimenForm

   a) Dempster-Shafer belief mass function is evaluated based on the node name similarities

TestAgent	**SpecimenAgent**
Test-TestResult=0.8	Specimen-Specimen=1.0
Control-TestControl=0.7	Form-SpecimenForm=0.4
Temperature-TestTemperature=0.7	Name-SpecimenName=0.25
Standard-TestStandard=0.65	Characterisation-SpecimenCharactesisation=.25

   b) Dempster-Shafer belief mass function is evaluated based on the node structure similarities Test(Control,Temperature,Standard)-

TestResult(TestControl, TestTemperature, TestStandard)= 0.5
Specimen(Name,Form,Characterization) and
Geometry(SpecimenForm,SpecimenName,
SpecimenChar)=0.6

c) Combined similarity, belief function can be calculated cooperatively by the two agents. TestResult in O_1 is similar concept to Test in O_2 with belief function 0.8 Geometry in O_1 is similar concept to Specimen in O_2 and Form in O_1 is similar property in SpecimenForm in O_2

5. New findings can be added to the broker knowledge:

$\forall$x TestAgent(x) and canAnswer(x,TestResult)

$\forall$x SpecimenAgent(x) and canAnswer(x,Geometry)

6. New knowledge is added to the blackboard

TestResult is similar to Test with belief function = 0.8
Control is similar to TestControl with belief function = 0.7
Temperature is similar to TestTemperature with belief function = 0.7
Standard is similar to TestStandard with belief function = 0.65
Specimen is similar to Specimen with belief function = 1.0
Form is similar to SpecimenForm with belief function = 0.4
Name is similar to SpecimenName with belief function = 0.25

The first identified problem my research addresses is establishing similarity mapping algorithms that make use of semantic similarity instead of string similarities. The considered approaches are as follows:

Based on the publications reviewed so far the following similarity measures have been discussed:

- Rule based similarity: A predefined set of rules gives hint of the similarity measure between concept and attributes. This method can be

cumbersome since the rules need to be set up by a human expert who has knowledge of the domain. Nevertheless this method can provide a really trustful similarity measure once in place.

- Content/Name and Meta learner: Solution is based on a different set of machine learning algorithms. The idea behind this concept is based on instances classifiers that can be can be trained on available data instances in order to assess similarity between the concept and/or attributes. The solution does not consider the hierarchical nature of concepts instead makes use of concepts and context in the instance representation.
- Sub-graph isomorphism approaches. Making use of the fact that ontologies are represented as hierarchical taxonomies, sub-graph isomorphism algorithms can be used to determine the similarities between concepts and their attributes. These solutions claims to achieve semantic matching between entities. Though as pointed out [47] these algorithms are computationally NP hard.

Combination of the above mentioned methods.

4.11 Agent communication protocol

Jade agents communicate through FIPA Agent Communication Language (ACL) messages. The structure of the standard ACL message is:

```
(query-ref
      :sender
      (agent-identifier
            :name broker@kmi.open.ac.uk
            :addresses (sequence iiop://foo.com/acc))
      :receiver (set
      (agent-identifier
            :name material@kmi.open.ac.uk
            :addresses (sequence iiop://foo.com/acc)))
      :language FIPA-SL
      :ontology FIPA-Ontol-Service-Ontology
      :content
      (iota ?level (ontol-relationship O1 O2 ?level)))
```

In JADE the ACL message content is a string as default but there are number of ways to use XML or application specific ontology to describe the communication. While ontology based communication seems promising at first sight the practical implementation includes several design assumptions which do not make it suitable for my prototype e.g. ontologies are only used to create the java classes that actually represents the concepts or predicates and message goes through several transformation during a particular communication action. Additionally every used ontology is a subset of the domain ontology or there exists a map between it and the domain ontology; the knowledge about these relationships (subset and mapping) is usually maintained by some ontology-dedicated agents.

In my prototype however mapping agents use SWI prolog engine to achieve reasoning capabilities so as a consequence I developed a simple XML based communication protocol (ACP) that is tightly integrated with the FOL formula representation and the specific nature of the question answering.

The two main entities are the query and the answer. The sub elements in each node depend on which agent communicates with whom e.g. the query and answer structure between the broker and the mapping agents is as follows:

```
<acp>
    <Query>
        <QueryFragment>hasIdentifier(Material,Cr Mo 10)</QueryFragment>
    </Query>
</acp>

<acp>
    <Answer>
        <Similarity>
            <Class ID="Material">
                <Source ID="Ontology 1" BMF="1">Material</Source>
                <Source ID="Ontology 2" BMF ="0.4">Subject</Source>
            </Class>
            <ObjectProperty ID="hasDetails">
                <Source ID="Ontology 1" BMF ="0.5">hasProductionDetails</Source>
                <Source ID="Ontology 2" BMF ="0.4">hasDesignation</Source>
                </ObjectProperty>
            <DataProperty ID="Identifier">
                <Source ID="Ontology 1" BMF ="0.6">Name</Source>
                <Source ID="Ontology 2" BMF ="1.0">Identifier</Source>
            </DataProperty>
        </Similarity>
    </Answer>
</acp>
```

4.12 Conclusion on pilot study

The pilot study described how we implemented a simple first version of the multi agent-mapping tool and discussed our preliminary ideas how uncertainty can be incorporated into the system. We evaluated our system using different versions of the same ontology (see appendices).

We have also outlined that incorporating belief mass function can improve the system correctness however this area has several open questions (see Research proposal) that if resolved can make the performance of the tool even better.

We intended to carry out a qualitative comparison of our pilot against other mapping tools e.g. GLUE [11] or InfoSleuth[15] but unfortunately after contacting several authors we cold not obtain a evaluation copy of their system. To compare

our proposed system against other approaches for discovering mappings between ontologies we rely completely on publications of these systems.

Automatic ontology or probabilistic mapping methods and employ a number of different techniques. For example, Prompt [18] algorithms compare graphs representing the ontologies or schemas, looking for similarities in the graph structure. GLUE [11] is an example of a system that employs machine-learning techniques to find mappings. GLUE uses multiple learners exploiting information in concept instances and taxonomic structure of ontologies. GLUE uses a probabilistic model to combine results of different learners. The before mentioned techniques are based mainly on linguistic analysis of concept names and natural-language definitions of concepts.

In the context of the Semantic Web, Ding [9] has proposed probabilistic extensions for OWL. In this model, the OWL language is extended to allow probabilistic specification of class descriptions. The authors then build a Bayesian Network based on this specification, which models whether or not an individual matches a class description and hence belongs to a particular class in the ontology.

Our implemented pilot study and our proposed research direction seems to be complementary to the before mentioned techniques for automatic, semi-automatic or probabilistic ontology mapping. Many of the above mentioned methods produced pairs of matching terms with some degree of certainty. We can use these results as input for creating belief function and combine these beliefs to improve the mapping produced by similarity algorithms or to suggest additional matches. In other words, our work complements and extends the work by other researchers in this area.

5. Research proposal

5.1 Proposed research issues

The introduction, motivation and research contributions are described in chapter 2.

As explained in the problem definition section (2.2) my PhD research issues fall into two distinctive but tightly correlated areas namely similarity mapping algorithms and probabilistic uncertainty handling and reasoning in a distributed environment for ontology mapping.

The main objective of my research is to investigate how probability theory as a means of assessing the likelihood of terms in different ontologies refer to the same or similar concepts can be harnessed in order to provide better answers to user queries in the context of question answering.

Further multi agent architectures for applying plausible reasoning to the problem of ontology mapping needs to be evaluated based on best practices for representing uncertain, incomplete, ambiguous, or controversial information in the Semantic Web. A part of the research will assess how probabilistic reasoning techniques applied to trust issues in the Semantic Web can be utilized in order to address the problem of distributed local information.

The following sections will explain in detail the proposed research issues:

5.2 Similarity mapping algorithms and measures in a distributed environment

It is acknowledged in the ontology research community that similarity mapping algorithms and measures can produce an average of 60-90 percent correct results. However nearly all of the proposed solutions suggest that

1. To assess similarity an algorithm or application has global knowledge of all information necessary to built up mapping.
2. Domain experts can validate the correctness of the mapping.

In the question answering context however we would like to investigate that if the knowledge or information that is necessary to built up a mapping is distributed among domain specific agents (each agent has information about a set of closely related concepts or group of concepts) than

- How the correctness of the mapping will be affected?
- What kind of similarity measures and distributed algorithms can improve the results that have been achieved with the current solutions?

The above-mentioned points are important and worth investigating because mapping the structured and distributed domain knowledge with ontologies within different communities is key to realizing real world Semantic Web applications. However, the decentralized nature of the Web makes this difficult, thus, hampering efficient knowledge sharing between them.

The need for interoperability and similarity mapping mechanisms between distributed ontologies are a key factor that needs to be investigated in an interactive and dynamic environment such as question answering in order to achieve mappings between distributed ontologies.

Our research objective is to investigate complex mappings and reasoning services about those mappings using uncertainty that we believe is necessary for comparing, combining ontologies, and for integrating data described using different ontologies.

Our further research objective is to investigate and provide a standardised approach for combining different simple similarity mapping techniques and to integrate it into a well established framework that would scale and outperform currently applied techniques in the field of ontology mapping and information integration.

As a starting point we consider SimilarityBase and SimilarityTop [43,44] algorithms investigated in the context of AQUA. We believe that these algorithms provide a promising direction towards a graph based concept and property

similarity matching. The main advantage of using these algorithms comes down to the fact that they use contextual neighborhood and evidential information about the arguments in the query. It defines a certain depth of concept hierarchy referred as the contextual neighborhood, which we believe is really intuitive comparing to the general idea of incorporating parent concepts into the query graph up to a common parent or root node in the concept hierarchy. In our prototype we implemented SimilarityBase and SimilarityTop algorithms. However we would like to extend these algorithms in a distributed environment and investigate how the usability and correctness of the matching results will be affected.

5.3 Role of distributed local knowledge in ontology mapping

Concerning the knowledge management perspective of our research it is based on the Distributed Knowledge Management (DKM) approach[48], in which subjective and social aspects of the real world are seriously taken into account.
Compared to the traditional Knowledge Management (KM) view of creating, codifying and disseminating knowledge as single, supposedly shared and objective classification we believe that Distributed Knowledge Management is the viable alternative to the concept of an existing "centralized or common knowledge" in the context of question answering on the WWW. The concept of Semantic Web is also based on the distributed knowledge idea, which is represented by different ontologies. However in the context of ontology mapping and information integration the current state of the art approaches does not reflect fully these ideas since as our literature review points out most of the mapping approaches are based on the centralized or common knowledge approach.
The reason we would like to investigate the distributed approach further in our research is because in complex environments like the question answering the knowledge is:

- Locally defined and based on the entities perspective i.e. subjective.

- Exchangeable between the local perspectives.

The basic argument is that knowledge cannot be viewed as a simple conceptualisation of the world, but it has to represent some degree of interpretation. Such interpretation depends on the context of the entities involved in the process.

This idea is rooted in the fact the different entities' interpretations are always subjective, since they occur according to an individual schema, which is than communicated to other individuals by a particular language. These schemas called mental spaces, contexts, or mental models have been investigated before[49,50,51].

As a consequence the local knowledge - which is a partial interpretations of the different concepts in the different domains - is represented by the different entities or within communities through a process of negotiating interpretations. As the literature review shows this process has not been fully been investigated to date and this fact serves as the main motivation why we pose this research question in our context.

By carrying out research in this area we expect that we can also examine if the limitations of the traditional knowledge management systems based on a single schema can also affect the future semantic web based applications. This is an important question, which affects the practical applicability of such systems since the consequences can involve gradual rejection of the system by the users.

5.4 Incorporating trust in the mapping process

In the Semantic Web, all kinds of information are expressed on a single information model framework, which allows us to connect different kinds of information from different sources and use them as a huge distributed database. In the present Semantic Web there is no mechanism for evaluating the trust of the particular sources.

Theoretically everyone can freely write and publish information on their Web page or in a database, which involves the possibility of incorporating incorrect information.
In the context of question answering the issue of trust in the different sources can significantly affect the overall system acceptance by the users thus the practical usability of the system. Imagine a scenario where the user is looking for existing experimental data for a finite element calculation. If the user can find numerous data sets but some of these data are incorrect or untrustworthy then there is a significant risk of the rejection of the whole system by the users.

There are lots of ways to define trust as the quantified belief by a trustor within a specified context.
Quantification reflects that a trustor can have various degrees of trust, which could be expressed as a numerical range or as a simple semantic classification. However a trust level for one context doesn't normally apply to a different context hence the attributes of trust depend on the trust context. As an example one can consider that in our multi agent framework the different specialised agents are able to assess similarity correctly only for part of the domain. Material agent knows everything about concepts related to material, specimen agent is for specimen etc.
However when specimen agent assesses similarity on information that belongs to material this similarity measure can negatively influence the overall result of the similarity assessment. Additionally the correctness of the source information also needs to be considered because test data from a material research institute can be reliable when it comes to material properties but not when specimen information is involved e.g. material chemical composition can be trusted from the material manufacturer but specimen surface heat treatment information can only be trusted from the laboratory who prepared the specimen.
Besides the above mentioned issues, determining initial trust values can be quite difficult, and the default values might be rather arbitrary and application dependent. In many practical situations, there might not be any past experience

for a specific trustee or context on which to base a trust evaluation. Thus the evaluation might have to depend on trust evaluation from a different context. There needs to exist some kind of trust evaluation between agents and sources.

The research questions that needs to be answered during this phase are:

- What factors need to be considered when expressing such initial trust?
- How to express these factors in a numerical way?
- How to maintain consistent trust factors between the agents and sources?
- How these factors can be effectively incorporated into a probabilistic framework?

In our multi agent framework we need to use the trust specification to influence the ontology mapping decisions and in combining evidence related to experience.

5.4 Converting similarity measures into belief masses

In this phase of the research we are trying to answer a single but very complex question:
Where will the belief masses will come from i.e. how can we use similarity measures to back our hypothesis and be able to make subjective judgment about the probabilities?
The subjective judgment or probability is the agent's actual judgment, normally representing what a human expert's judgment would be, in view of his information to date and of his sense of other people’s information, even if the particular judgment is not shared by the other experts.
The question makes sense because our similarity algorithms can provide numerical values between certain types of concept name or property matches however these numbers need to be translated into a quantitative form of belief.
This is not a straightforward question and several studies tried to investigate how probability can be assessed from similarities [53,54,55].

As the literature review showed researchers have begun investigating the Baysian theorem as a probabilistic framework for handling uncertainty in the context of ontology mapping. The reason why we choose to investigate Dempster-Shafer theory in our multi agent framework for ontology mapping is because we believe that our approach will prove to be more effective. We base our belief on the fact that according to the behavioral theory the human reasoning about probability rarely follows Bayes's Theorem[52]. In our research we need to establish probability statements under this concept to represent the degree of rational belief that the agent holds about the likelihood of correctness of the mapping.

Investigating this kind of subjective concept of probability enables us to assess the usability of probability statements about the mappings even if these are non-repeatable events, and provides a mechanism for formulating and understanding practical beliefs about probabilities.

Further we will investigate the use of heuristics in the estimations of likelihood of uncertain events. The heuristics, of which representativeness and availability can make estimation of probabilities computationally tractable, but often result in biases (violations of the probability axioms).

The representativeness of heuristics can influence the judgments of the probability that an object belongs to a category is based on the similarity between the object and the category prototype e.g. Fracture-Time is actually the same as Time at fracture if it in the context of a specimen.

We believe that these heuristics can significantly influence the outcome of our mapping process therefore it should be investigated during our research.

5.5 Algorithms for variable elimination sequence in a distributed environment

One of the main difficulties applying Dempster-Shafer framework is its computational complexity with complex domains.

Complex mean a large number of variables or hypothesizes that needs to be combined by the system. In our ontology mapping context for question answering we clearly face the problem that the computation quickly become infeasible when we need to combine evidence even if we divide our domain between specialised agents. This can lead to a performance breakdown so any practical implementation will be deemed to rejection by the users.

To resolve this problem we need to investigate the possibility of a reasonable optimalisation method.

Dividing our domain between the specialised agents is the first step that can reduce the size of the state space however we need to investigate if constrains in the Dempster-Shafer theory e.g. hypothesises are mutually exclusive are still valid in our optimalised environment.

To carry out inference under uncertainty we are going to use a valuation network that uses joint trees(undirected graphs) with a message passage scheme. It is clear that different valuation elimination sequences will lead to different joint tree structures. In our research we will examine how different variable elimination sequence algorithms influence the structure of our joint tree with respect to the fact that this graph should be distributed between the specialised agents. The variable elimination problem is well known from the field of dynamic programming where it is used to transforms the problem into an equivalent one, having less variables so we can decrease the computational complexity.

These approaches typically combine the solutions to sub-problems, which are not independent of each other.

However it is well known that the performance of elimination algorithms is likely to suffer from the exponential space and time necessary to calculate the solution.

The main motivation of our research in this direction is to find optimal elimination algorithms that satisfies the basic dynamic programming conditions:

1. Creates optimal graph substructures i.e. an optimal solution that must involve optimal solutions to its sub-problems.

2. Sub-problems must overlap, i.e. the recursive algorithm considers the same problems on different occasions rather than creating new ones every time.

In this phase of the research we will also investigate the performance of different elimination algorithms and the usability of these algorithms in our scenario.

5.6 Algorithms for distributed valuation network optimalizations

Local computation works with variable elimination. This leads in a very straightforward way to a graphical structure called join trees. The computations on join trees themselves are organized as a message-passing algorithm and there exists different architecture types taking advantage of additional properties of the underlying valuation algebra.

This solution perfectly fit into our scenario however as literature review showed no research has been carried out on the applicability of a valuation network in a distributed environment. In our scenario each agent is associated with an environment or a problem domain of interest and carries a model or a representation or some prior knowledge about the domain. However multiple agents must collectively reason about the state of the domain based on their local knowledge, local observation, and limited communication. This will imply that in our question answering scenario where one query contains answers from multiple sources the before mentioned joint tree that represent the knowledge must be distributed among different agents.

A number of different algorithms exist for constructing the junction tree based on cluster size, cluster weight, fill in size, fill in weight and total weight (optimal). However distributed or parallel approaches has not fully been investigated yet. Additionally to date all methods in the AI community of finding a joint tree has no guarantee of performance and could perform differently when applied to a particular problem.

Our research motivation is to address the above mentioned problem and investigate how to connect the different join trees together so that only one single join tree is finally obtained so the reasoning can be carried out effectively. In this phase of the research we would like to answer the following questions:

Should agents exchange their observations or their beliefs?

- If each agent has only a partial perspective of the domain, what should be the relationship between their beliefs?
- should agents be allowed to hold inconsistent beliefs?
- Is there such a thing as the collective belief of multiple agents and can it be represented with valuation network?

In order to answer these questions we need to assess different joint tree construction algorithms and investigate the limitations in a distributed environment.

5.7 Work plan

The research will be carried out between November 2005 and November 2007, giving a total of 24 month as depicted in figure 14. The Gantt chart clearly shows which subtasks are dependent on other subtasks and parallel activities are also represented. The time scales include research plus system implementation. The detailed activities are as follows:

1. Similarity mapping algorithms and measures in a distributed environment (duration 60 days): First we need to assess the possibility of using similarity measure combination in a distributed environment. This task is based on our experiments carried out in our prototype. Once we utilized our similarity measures and its combination we need to adapt the SimilarityBase and SimilarityTop algorithms into this distributed context. The outcome of this phase is a qualitative comparison of the

methodologies, which highlight the improvements that can be achieved comparing to the “traditional” single application concepts.

2. Role of distributed local knowledge in ontology mapping (duration 50 days): In this phase of the research we will develop and enrich the specialized agents local knowledge base and compare the mapping results with the outcome of the phase one. Further we will compare, which information needs to be shared or exchanged between to agents in order to increase the correctness of the mappings. The outcome of this phase is an analysis of the effects of using localized knowledge in the context of ontology mapping.
3. Incorporation trust in the mapping process (duration 30 days): This activity runs partly parallel with activity two. In this phase of the research we will express the concept of trust in a numerical way and apply it to our similarity measures. The outcome of this activity is to assess how trust can influence the mapping results and how it can be maintained in our distributed environment.
4. Converting similarity measures into belief masses (duration 60 days): In this phase of the research we will establish a methodology which describes how the belief can be deducted from the similarity measures produced by the earlier phases of the research. This is a key activity that will clarify how numerical belief function can be produced and used in our context. We will investigate different heuristics and its effect on the estimations of likelihood of uncertain events.
5. Algorithms for variable elimination sequence in a distributed environment (duration 100 days): One of the most difficult tasks is how belief function combination can be made feasible in a complex domain with a large number of variables. In this phase of the research we will investigate how a graph based optimalisation method can be adapted into our distributed scenario namely how variable elimination can affect the optimal graph substructures that will be used to calculate the combined belief over the state space. The outcome of this phase is a performance assessment of

different elimination algorithms and the usability of these algorithms in our context.

6. Algorithms for distributed valuation network optimalizations (duration 100 days): This phase further investigates the results of the previous activity and assesses the applicability of each agent ‘s partial belief perspective of the domain. Optimalisation algorithms will be developed and compared with each other based on their performance and applicability in a distributed environment. The key objective is to establish effective reasoning with representing the knowledge that is distributed among different agents.
7. Writing dissertation (duration 132 days): In the last phase of the research the dissertation will be compiled based on the results achieved during the earlier phases. The 132 days planned here is deliberately pessimistic to provide some extra time to enable us to compensate for over-runs in the previous phases.

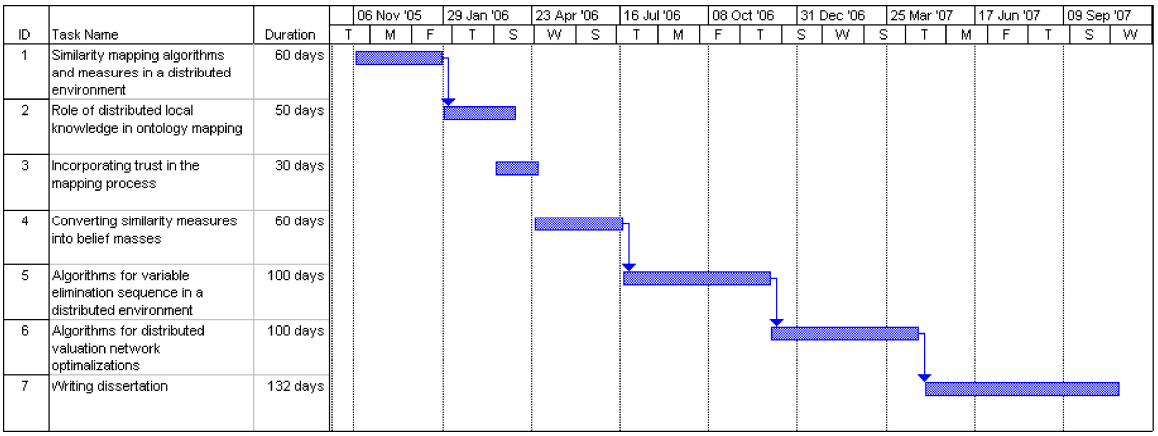


Figure 14.

6. References

[1] SHOE Homepage (2001). http://www.cs.umd.edu/projects/plus/SHOE/ SHOE

[2] RDF Homepage (2004). http://www.w3.org/RDF/ W3C

[4] DAML Homepage (2004). http://www.daml.org/

[5] OIL Homepage (2003) http://www.ontoknowledge.org/oil/

[6] DAML+OIL Homepage (2003). http://www.daml.org/2001/03/daml+oil-index

[7] OWL Guide Version 1.0 Draft (2004). http://www.w3.org/TR/owl-guide/

[8] Ehrig M., Sure Y. (2004). Ontology Mapping-An Integrated Approach, Proceedings of the First European Semantic Web Symposium, volume 3053 of Lecture Notes in Computer Science, pp. 76-91. Springer Verlag, Heraklion, Greece.

[9] Ding Z., Peng Y. (2004). A probabilistic extension to ontology language owl. In Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37), Big Island, Hawaii.

[10] Swap project homepage (2002). http://swap.semanticweb.org

[11] Doan A. H., Madhavan J., Domingos P., Halevy A. (2002). A. Learning to Map between Ontologies on the Semantic Web. In Proceedings of 11th International World Wide Web Conference (WWW2002), Honolulu, Hawaii.

[12] Doan A. H., Domingos P., Halevy A. (2001). A. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In Proceedings of ACM SIGMOD Conference(SIGMOD 2001), Santa Barbara, USA.

[13] Garcia-Molina H., et al. (1997). The TSIMMIS Approach to Mediation: Data Models and Languages, Journal of Intelligent Information Systems, 8(2):117-132.

[14] Halevy A. (1998). The Information manifold approach to data integration, IEEE Intelligent Systems, 1312-16.

[15] Bayardo R., et al.. Infosleuth (1997). Agent-based Semantic Integration of Information in Open and Dynamic Environments. In Proceedings of ACM SIGMOD Conference on Management of Data, 195-206. Tucson, Arizona.

[16] Beneventano D., Bergamaschi S., Guerra F., Vincini M. (2001). The MOMIS Approach to Information Integration. In Proceedings of 3rd International Conference on Enterprise Information Systems (ICEIS), 194-198, Setúbal, Portugal.

[17] Stumme G., Madche A. (2001). FCA-Merge: Bottom-up Merging of Ontologies. In Proceedings of 7th Intl. Conf. on Artificial Intelligence (IJCAI '01), 225-230. Seattle, US.

[18] Noy N. F., Musen M. A. (2000). PROMPT Algorithm and Tool for Automated Ontology Merging and Alignment In the Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). Austin, US.

[19] Maedche A., Motik B., Silva N., Volz R. (2002). Mafra - a mapping framework for distributed ontologies. In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Siguenza, Spain.

[20] Dempster A.P. (1968). A generalization of Bayesian inference. Journal of the Royal Statistical Society, Series B 30 205-247.

[21] Shafer G. (1976). A Mathematical Theory of Evidence. Princeton University Press., Princeton , US.

[22] Smets Ph. (2001). Decision Making in a Context where Uncertainty is Represented by Belief Functions, In Belief Functions in Business Decisions. Physica-Verlag, 17-61, Heidelberg, Germany.

[23] Smets Ph., Hsia Y.-T. (1991). Default Reasoning and the Transferable Belief Model, In Uncertainty in Artificial Intelligence 6, Elsevier Science Publishers, 495-504.

[24] Levi I. (1983). Consonance, dissonance and evidentiary mechanisms. In Gardenfors P., Hansson B. and Sahlin N.E. (eds) Evidentiary value: philosophical, judicial and psychological aspects of a theory. C.W.K. Gleerups, 27-43, Lund, Sweden.

[25] Voorbraak F. (1989). A computationally efficient approximation of Dempster-Shafer theory. International Journal of Machine Studies, 30:525–536.

[26] Lowrance J., Garvey T., Strat T. (1986). A framework for evidential-reasoning systems. In Proceedings of the 8th National Conference of the American Association for Artificial Intelligence, pages 896–903. Boston, US.

[27] Tessem B. (1993). Approximations for efficient computation in the theory of evidence. In Artificial Intelligence, 61:315–329.

[28] Bauer M. (1996). Approximations for decision making in the Dempster-Shafer theory. In Uncertainty in Artificial Intelligence, pages 339–344.

[29] Kreinovich V., Bernat A., Borrett W., Villa E. (1994). Monte-carlo methods make Dempster-Shafer formalism feasible. In Advances in the Dempster-Shafer Theory of Evidence, pages 175–191.

[30] Wilson N. (1991). A monte-carlo algorithm for dempster-shafer belief. Technical Report QMWDCS-1991-535, Queen Mary College, Department of Computer Science, US.

[31] Yager R. R., Kacprzyk J., Fedrizzi M. (1994). Advances in the Dempster-Shafer Theory of Evidence, John Wiley & Sons, Inc. Wiley, New York,US.

[32] Thoma H. M. (1991). Belief function computations. In Goodman, I. R., Gupta M. M., Nguyen H. T., and Rogers G. S., editors, Conditional Logic in Expert Systems, pages 269–308. North-Holland, Amsterdam.

[33] Kennes R., Smets P. (1990). Computational aspects of the Moebius transform. In Uncertainty in Artificial Intelligence 6, pages 401-416.

[34] Henrion M., Shachter R. D., Kanal L. N., Lemmer J. F. (1990). Uncertainty in Artificial Intelligence 5. North Holland, Amsteram.

[35] Shafer G., Shenoy P. P., Mellouli K. (1987). Propagating belief functions in qualitative markov trees. Int. Journal. Approx. Reasoning, 1:349–400.

[36] Arnborg S., Corneil D. G., Proskurowski A. (1987). Complexity of finding embeddings in a k-tree. In SIAM Journal of Algebraic and Discrete Methods, volume 8, pages 277–284.

[36] Lehmann N. (2001). Argumentation System and Belief Functions. PhD thesis, Department of Informatics, University of Fribourg, Switzerland.

[37] Shenoy P. (1997). Binary join trees for computing marginals in the shenoy-shafer architecture, International Journal of Approximate Reasoning, Vol. 17, 239-263.

[38] Lauritzen S. L., Spiegelhalter D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion), Journal of Royal Statistical Society, Series B, 50(2), 157–224.

[39] Jensen F. V., Lauritzen S. L., Olesen K. G. (1990b). Bayesian updating in causal probabilistic networks by local computation, Computational Statistics Quarterly, 4:269-282.

[40] Bissig R., Kohlas J., Lehmann N. (1997). Fast-division architecture for Dempster-Shafer belief functions, In Proc. of the International Joint Conference on Qualitative and Quantitative Practical Reasoning (ECSQARU/FAPR-97), Lecture Notes in Artificial Intelligence, 198-209.

[41] Parsons P., Hunter A. (1998). A Review of Uncertainty Handling Formalisms, Applications of Uncertainty Formalisms, Lecture Notes in Computer Science, 8-37.

[42] Walley P. (1991). Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London, UK.

[43] Vargas-Vera M., Motta E. (2004). AQUA - Ontology-based Question Answering System. Third International Mexican Conference on Artificial Intelligence (MICAI-2004), Lecture Notes in Computer Science 2972 Springer Verlag. Eds R. Monroy, G. Arroyo-Figueroa, L.E. Sucar and J.H. Sossa Azuela. pp. 468-477, Mexico City, Mexico.

[44] Vargas-Vera M., Motta E., Domingue, J. (2003). AQUA: An Ontology-Driven Question Answering System. AAAI Spring Symposium, New Directions in Question Answering, Stanford University, 53-57.

[45] Vargas-Vera M., Motta E. (2004). A Knowledge-Based Approach to Ontologies Data Integration. KMi-TR-152, The Open University, July 2004,UK.

[46] Vargas-Vera M., Motta M. (2004). An Ontology-driven Similarity Algorithm. KMI-TR-151, Knowledge Media Institute, The Open University, July 2004. UK.

[47] Atallah M. J. (1999). Algorithms and Theory of Computation Handbook, ed., CRC Press LLC.

[48] Bonifacio M., Bouquet P., Traverso P. (2002). Enabling Distributed Knowledge Management: Managerial and Technological Implications . Technical Report DIT-02-069, Informatica e Telecomunicazioni, University of Trento, Italy.

[49] Fauconnier G. (1985). Mental Spaces: Aspects of Meaning Construction in Natural Language , Bradford Books, MIT Press,US.

[50] Giunchiglia F., Ghidini C. (2000). Local Models Semantics, or Contextual Reasoing = Locality + Compatibility, Artificial Intelligence, 127(2), p. 221–259.

[51] Johnson-Laird P. (1992). Mental Models. Cambridge University Press , Cambridge,UK.

[52] Kahneman D. et al. (1982). Judgment under uncertainty: Heuristics and Biases ,Conservatism in Human Information Processing,359, 361–69.

[53] Blok S., Medin D.L., Osherson D. (2003). Probability from similarity. In Proceedings of AAAI Conference on Commonsense reasoning , Stanford University, US.

[54] Ding C.H. (1999). A Similarity-based probability model for Latent Semantic Indexing. In Proceedings of SIGIR'99, volume 1, pages 58-65.

[55] Juslin P., Nilsson H., Olsson H. (2001). Where Do Probability Judgments Come From? Evidence for Similarity–Graded Probability, In Proceedings of 23rd Annual conference of the cognitive science society, Edinburgh, Scotland

[56] Finin T., et al. (2005). Swoogle: Searching for knowledge on the Semantic Web, In Proceedings of Twentieth National Conference on Artificial Intelligence, (AAAI 05), Pittsburgh, Pennsylvania

[57] Yen J. (1989). GERTIS: A Dempster-Shafer Approach to Diagnosing Hierarchical Hypotheses. Commun. ACM 32(5): 573-585.

[58] Monge A. E., Elkan C. P. (1996). The field-matching problem: algorithm and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, US.

[59] Cohen W., Ravikumar P., Fienberg, S. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks, In Proceedings of Information Integration on the Web (IIWeb 2003), Accapulco, Mexico.

[60] Falcone R., Singh M. P., Tan Y. (2001). Trust in Cyber-societies, Integrating the Human and Artificial Perspectives, Springer.-Lecture Notes in Computer Science, Vol. 2246.

[61] Ramchurn S. D., Huynh D., Jennings N. R. (2004). Trust in multiagent systems, In the Knowledge Engineering Review 19 (1) 1-25.

[62] Neuman J., Morgenstern O. (1944). The Theory of Games and Economic Behaviour. Princeton, NJ: Princeton University Press, US.

[63] Sabater J., Sierra C. (2002). REGRET: a reputation model for gregarious societies. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 475–482.

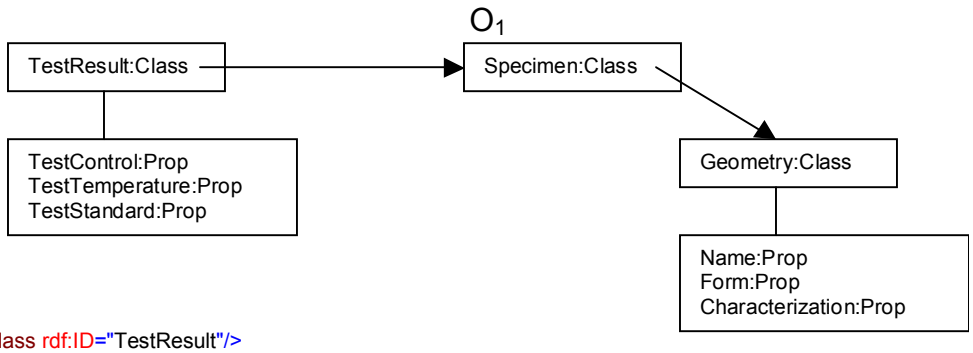
[64] Buskens V. (1998). The social structure of trust. In Social Networks 20, 265–298.

[65] Wooldridge M. (2002). An Introduction to MultiAgent Systems. Chichester: John Wiley and Sons.

[66] Sandholm T. (1999). Distributed rational decision-making. In Weiss, G. (ed.), Multi-Agent Systems: A Modern Approach To Distributed Artificial Intelligence. MIT Press, US.

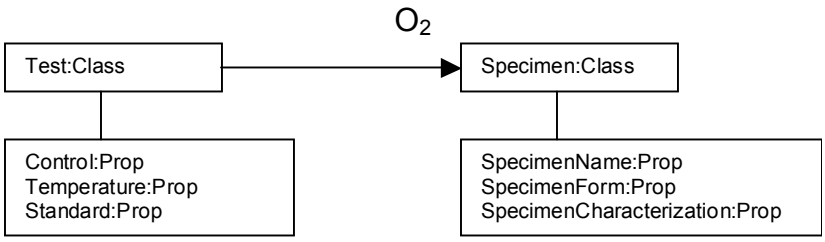
[67] Poslad S., Calisti M., Charlton P. (2002). Specifying standard security mechanisms in multi-agent systems. In Workshop on Deception, Fraud and Trust in Agent Societies, AAMAS 2002, 122–127, Bologna, Italy.

7. Appendicies




```
<owl:Class rdf:ID="TestResult"/>
<owl:DatatypeProperty rdf:ID="TestControl">
   <rdfs:domain rdf:resource="# TestResult "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="TestTemperature">
   <rdfs:domain rdf:resource="# TestResult "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="TestStandard">
   <rdfs:domain rdf:resource="# TestResult "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:ObjectProperty rdf:ID="hasSpecimen">
   <rdfs:range rdf:resource="#Test"/>
   <rdfs:domain rdf:resource="#Specimen"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Specimen"/>
 <owl:Class rdf:ID="Geometry"/>
 <owl:ObjectProperty rdf:ID="hasGeometry">
   <rdfs:range rdf:resource="#Specimen"/>
   <rdfs:domain rdf:resource="#Geometry"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="Name">
   <rdfs:domain rdf:resource="#Geometry "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="Form">
   <rdfs:domain rdf:resource="#Geometry "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="Characterization">
   <rdfs:domain rdf:resource="#Geometry "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>

<Test>
  <hasSpecimen>
    <Specimen>
      <Geometry>
      <Name>vlmi</Name>
      <Form>bar</Form>
      <Characterization>vlmi</Characterization>
      </Geometry>
    </Specimen>
  </hasSpecimen>
    <TestControl>strain-controlled</TestControl>
    <TestTemperature>560</TestTemperature>
    <TestStandard>ISO 9001</TestStandard>
 </Test>
```




```
<owl:Class rdf:ID="Test"/>
<owl:DatatypeProperty rdf:ID="Control">
   <rdfs:domain rdf:resource="# Test"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Temperature">
   <rdfs:domain rdf:resource="# Test"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="Standard">
   <rdfs:domain rdf:resource="# Test"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:ObjectProperty rdf:ID="hasSpecimen">
   <rdfs:range rdf:resource="#Test"/>
   <rdfs:domain rdf:resource="#Specimen"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="Specimen"/>
 <owl:Class rdf:ID="Geometry"/>
<owl:DatatypeProperty rdf:ID="SpecimenName">
   <rdfs:domain rdf:resource="#Specimen "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="SpecimenForm">
   <rdfs:domain rdf:resource="#Specimen "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="SpecimenCharacterization">
   <rdfs:domain rdf:resource="#Specimen "/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>

<Test>
  <hasSpecimen>
    <Specimen>
      <SpecimenName>vlmi</SpecimenName>
      <SpecimenForm>bar</ SpecimenForm>
      <SpecimenCharacterization>vlmi</SpecimenCharacterization>
    </Specimen>
  </hasSpecimen>
    <Control>strain-controlled</Control>
    <Temperature>560</Temperature>
    <Standard>ISO 9001</Standard>
 </Test>
```

System snapshot in JADE